

## Model Railroad System

2.2.1

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Internals (developer) documentation.</b>	<b>1</b>
1.1	Introduction . . . . .	1
<b>2</b>	<b>Module Index</b>	<b>3</b>
2.1	Modules . . . . .	3
<b>3</b>	<b>Namespace Index</b>	<b>5</b>
3.1	Namespace List . . . . .	5
<b>4</b>	<b>Hierarchical Index</b>	<b>7</b>
4.1	Class Hierarchy . . . . .	7
<b>5</b>	<b>Class Index</b>	<b>13</b>
5.1	Class List . . . . .	13
<b>6</b>	<b>File Index</b>	<b>21</b>
6.1	File List . . . . .	21

<b>7</b>	<b>Module Documentation</b>	<b>23</b>
7.1	Azatrax	23
7.1.1	Detailed Description	23
7.2	FCFSupportModule	24
7.3	ParserClasses	25
7.3.1	Detailed Description	25
7.3.2	Function Documentation	25
7.3.2.1	tcl_socketpair()	25
7.4	TimeTableSystem	26
7.4.1	Detailed Description	26
7.5	TimeTableSystemTcl	27
7.5.1	Detailed Description	27
7.5.2	Function Documentation	27
7.5.2.1	ForEveryCab()	28
7.5.2.2	ForEveryNote()	28
7.5.2.3	ForEveryPrintOption()	29
7.5.2.4	ForEveryStation()	29
7.5.2.5	ForEveryTrain()	30
7.5.2.6	NewCreateTimeTable()	30
7.5.2.7	OldCreateTimeTable()	31
7.5.2.8	TT_ListToStringListString()	31
7.5.2.9	TT_StringListToList()	31
7.5.3	Variable Documentation	33
7.5.3.1	Tcl_Result	33
7.6	Station	34
7.6.1	Detailed Description	34
7.6.2	Typedef Documentation	34
7.6.2.1	OccupiedMap	35

7.6.2.2	StationVector	35
7.6.2.3	StorageTrackMap	35
7.7	Train and support classes.	36
7.7.1	Detailed Description	36
7.7.2	Typedef Documentation	36
7.7.2.1	StopVector	36
7.7.2.2	TrainNumberMap	36
7.8	Cab	37
7.8.1	Detailed Description	37
7.8.2	Typedef Documentation	37
7.8.2.1	CabNameMap	37
7.9	TclSocketCANModule	38
7.9.1	Detailed Description	38
7.9.2	Function Documentation	38
7.9.2.1	SocketCAN()	38
7.9.2.2	Tclsocketcan_SafeInit()	39
7.9.3	Variable Documentation	39
7.9.3.1	i	39
7.9.3.2	TclSocketCAN	39
7.10	TclwiringpiModule	40
7.11	Cmri	41
7.11.1	Detailed Description	41
7.12	CTIAcela	42
7.12.1	Detailed Description	45
7.12.2	Typedef Documentation	45
7.12.2.1	addresstype	45
7.12.2.2	filterthreshtype	45
7.12.2.3	momtype	45

7.12.2.4	speedtype	46
7.12.2.5	ubyte	46
7.12.3	Function Documentation	46
7.12.3.1	_handleSRQ()	46
7.12.3.2	_readbyte()	46
7.12.3.3	_readevent()	47
7.12.3.4	_transmit()	47
7.12.3.5	Activate()	47
7.12.3.6	Blink()	48
7.12.3.7	ConfigureSensor()	48
7.12.3.8	Control16()	48
7.12.3.9	Control4()	49
7.12.3.10	Control8()	50
7.12.3.11	CTIAcela()	50
7.12.3.12	Deactive()	51
7.12.3.13	EmergencyStop()	51
7.12.3.14	HaveData()	51
7.12.3.15	highbyte()	51
7.12.3.16	lowbyte()	52
7.12.3.17	NetworkOffline()	52
7.12.3.18	NetworkOnline()	52
7.12.3.19	OnlineP()	53
7.12.3.20	pack4()	53
7.12.3.21	pack8()	53
7.12.3.22	Poll()	54
7.12.3.23	PulseOff()	54
7.12.3.24	PulseOn()	55
7.12.3.25	Query()	55

7.12.3.26 Read()	55
7.12.3.27 Read16()	56
7.12.3.28 Read4()	56
7.12.3.29 Read8()	56
7.12.3.30 ReadAll()	57
7.12.3.31 ReadRevision()	57
7.12.3.32 ResetNetwork()	57
7.12.3.33 ReverseBlink()	57
7.12.3.34 Signal2()	58
7.12.3.35 Signal3()	58
7.12.3.36 Signal4()	59
7.12.3.37 SignalBrightness()	59
7.12.3.38 SignalSettings()	59
7.12.3.39 SRQControl()	60
7.12.3.40 Throttle()	60
7.12.3.41 validate()	60
7.12.3.42 ~CTIAcela()	61
7.12.4 Variable Documentation	61
7.12.4.1 _timeout	61
7.12.4.2 CTI_DeviceMap	61
7.12.4.3 ctiacula	61
7.12.4.4 dataavailable	61
7.12.4.5 FilterSelectBits	62
7.12.4.6 LampBits	62
7.12.4.7 maxtries	62
7.12.4.8 networkonline	62
7.12.4.9 Opcodes	62
7.12.4.10 Responses	62

---

7.12.4.11 ttyfd . . . . .	62
7.13 LCCModule . . . . .	63
7.13.1 Detailed Description . . . . .	63
7.14 XPressNetModule . . . . .	64
7.14.1 Detailed Description . . . . .	64
7.15 NCEModule . . . . .	65
7.15.1 Detailed Description . . . . .	65
7.16 TclCommon . . . . .	66
7.16.1 Detailed Description . . . . .	66
7.17 LinuxGpio . . . . .	67
7.17.1 Detailed Description . . . . .	67
7.18 Cmri Support code . . . . .	68
7.18.1 Detailed Description . . . . .	68
7.19 RaildriverClientModule . . . . .	69
7.19.1 Detailed Description . . . . .	69
7.20 Graphics Support Code . . . . .	70
7.21 GRSupportModule . . . . .	71
7.21.1 Detailed Description . . . . .	71
7.22 SatelliteModule . . . . .	72
7.22.1 Detailed Description . . . . .	72
7.23 FCFSupport . . . . .	73
7.23.1 Detailed Description . . . . .	73

---



<b>8 Namespace Documentation</b>	<b>75</b>
8.1 azatrax Namespace Reference	75
8.1.1 Detailed Description	75
8.1.2 Tcl Package Provided	75
8.1.3 Library Provided	76
8.2 CabWidgets Namespace Reference	76
8.2.1 Detailed Description	76
8.2.2 Package provided	76
8.3 cmri Namespace Reference	76
8.3.1 Detailed Description	77
8.3.2 Package provided	77
8.3.3 Typedef Documentation	77
8.3.3.1 ByteList	78
8.3.3.2 uatype	78
8.3.3.3 ubyte	78
8.3.4 Enumeration Type Documentation	78
8.3.4.1 CardType	78
8.4 CmriSupport Namespace Reference	79
8.4.1 Detailed Description	79
8.4.2 Package provided	79
8.5 CTCPanel Namespace Reference	79
8.5.1 Detailed Description	81
8.5.2 Package provided	81
8.5.3 Function Documentation	81
8.5.3.1 leverMethods()	81
8.5.3.2 standardMethods()	81
8.5.3.3 trackworkmethods()	82
8.5.3.4 verifyBoolMethod()	82

8.5.3.5	<a href="#">verifyColorMethod()</a>	82
8.5.3.6	<a href="#">verifyDoubleMethod()</a>	82
8.5.3.7	<a href="#">verifyOrientation8Method()</a>	82
8.5.3.8	<a href="#">verifyPositionMethod()</a>	83
8.6	<a href="#">ctiacela Namespace Reference</a>	83
8.6.1	<a href="#">Detailed Description</a>	83
8.6.2	<a href="#">Package provided</a>	84
8.7	<a href="#">FCFSupport Namespace Reference</a>	84
8.7.1	<a href="#">Detailed Description</a>	86
8.7.2	<a href="#">Tcl Package Provided</a>	86
8.7.3	<a href="#">Library Provided</a>	86
8.7.4	<a href="#">Typedef Documentation</a>	86
8.7.4.1	<a href="#">CarTypeMap</a>	86
8.7.4.2	<a href="#">CarTypeOrderVector</a>	86
8.7.4.3	<a href="#">CarVector</a>	87
8.7.4.4	<a href="#">DivisionMap</a>	87
8.7.4.5	<a href="#">DivisionSymbolMap</a>	87
8.7.4.6	<a href="#">DivisionVector</a>	87
8.7.4.7	<a href="#">IndustryMap</a>	87
8.7.4.8	<a href="#">IndustryVector</a>	87
8.7.4.9	<a href="#">OwnerMap</a>	88
8.7.4.10	<a href="#">StationMap</a>	88
8.7.4.11	<a href="#">StationVector</a>	88
8.7.4.12	<a href="#">stringVector</a>	88
8.7.4.13	<a href="#">SwitchListElementVector</a>	88
8.7.4.14	<a href="#">TrainMap</a>	89
8.7.4.15	<a href="#">TrainNameMap</a>	89
8.7.5	<a href="#">Function Documentation</a>	89

8.7.5.1	<a href="#">operator&lt;&lt;() [1/2]</a>	89
8.7.5.2	<a href="#">operator&lt;&lt;() [2/2]</a>	89
8.8	<a href="#">FCFSupport::PDFFileStructures Namespace Reference</a>	90
8.8.1	<a href="#">Detailed Description</a>	91
8.8.2	<a href="#">Typedef Documentation</a>	91
8.8.2.1	<a href="#">NamedIndirectObjectMap</a>	92
8.8.2.2	<a href="#">PageLabelDictionaryNumMap</a>	92
8.8.2.3	<a href="#">PageLabelTreeKidVector</a>	92
8.8.2.4	<a href="#">PDFStreamVector</a>	92
8.8.3	<a href="#">Function Documentation</a>	92
8.8.3.1	<a href="#">QuotePDFString()</a>	92
8.9	<a href="#">FileEntry Namespace Reference</a>	93
8.9.1	<a href="#">Detailed Description</a>	93
8.9.2	<a href="#">Package provided</a>	94
8.9.3	<a href="#">Function Documentation</a>	95
8.9.3.1	<a href="#">_destroy()</a>	95
8.9.3.2	<a href="#">_openFile()</a>	95
8.9.3.3	<a href="#">_path_command()</a>	95
8.9.3.4	<a href="#">bind()</a>	96
8.9.3.5	<a href="#">cget()</a>	96
8.9.3.6	<a href="#">configure()</a>	96
8.9.3.7	<a href="#">create()</a>	97
8.10	<a href="#">gettext Namespace Reference</a>	97
8.10.1	<a href="#">Detailed Description</a>	97
8.10.2	<a href="#">Package provided</a>	98
8.10.3	<a href="#">Function Documentation</a>	98
8.10.3.1	<a href="#">_()</a>	98
8.10.3.2	<a href="#">_m()</a>	98

8.10.3.3	<code>_mx()</code>	98
8.11	GRSupport Namespace Reference	98
8.11.1	Detailed Description	99
8.11.2	Package provided	99
8.11.3	Function Documentation	99
8.11.3.1	<code>_ROPI()</code>	100
8.11.3.2	<code>_ROPI2()</code>	100
8.11.3.3	<code>DegreesToRadians()</code>	100
8.11.3.4	<code>RadiansToDegrees()</code>	100
8.11.3.5	<code>VerifyBooleanMethod()</code>	101
8.11.3.6	<code>VerifyColorMethod()</code>	101
8.11.3.7	<code>VerifyDoubleMethod()</code>	101
8.11.3.8	<code>VerifyIntegerMethod()</code>	101
8.11.3.9	<code>VerifyOrientationHVMMethod()</code>	101
8.11.4	Variable Documentation	101
8.11.4.1	<code>PI</code>	102
8.11.4.2	<code>PI2</code>	102
8.12	HTMLHelp Namespace Reference	102
8.12.1	Detailed Description	102
8.12.2	Package provided	102
8.13	Instruments Namespace Reference	103
8.13.1	Detailed Description	103
8.13.2	Package provided	103
8.13.3	Function Documentation	103
8.13.3.1	<code>CommonOptions()</code>	103
8.14	LabelComboBox Namespace Reference	104
8.14.1	Detailed Description	105
8.14.2	Package provided	105

8.14.3	Function Documentation	106
8.14.3.1	<code>_destroy()</code>	106
8.14.3.2	<code>_path_command()</code>	106
8.14.3.3	<code>bind()</code>	106
8.14.3.4	<code>cget()</code>	107
8.14.3.5	<code>configure()</code>	107
8.14.3.6	<code>create()</code>	107
8.14.3.7	<code>get()</code>	108
8.14.3.8	<code>getlistbox()</code>	108
8.14.3.9	<code>getvalue()</code>	108
8.14.3.10	<code>icursor()</code>	109
8.14.3.11	<code>post()</code>	109
8.14.3.12	<code>setvalue()</code>	109
8.14.3.13	<code>unpost()</code>	110
8.15	LabelSelectColor Namespace Reference	110
8.15.1	Detailed Description	110
8.15.2	Package provided	111
8.15.3	Function Documentation	111
8.15.3.1	<code>_destroy()</code>	111
8.15.3.2	<code>_path_command()</code>	112
8.15.3.3	<code>cget()</code>	112
8.15.3.4	<code>ColorPopup()</code>	112
8.15.3.5	<code>configure()</code>	114
8.15.3.6	<code>create()</code>	114
8.16	LabelSpinBox Namespace Reference	114
8.16.1	Detailed Description	115
8.16.2	Package provided	116
8.16.3	Function Documentation	116

8.16.3.1	<code>_destroy()</code>	116
8.16.3.2	<code>_path_command()</code>	116
8.16.3.3	<code>bind()</code>	117
8.16.3.4	<code>cget()</code>	117
8.16.3.5	<code>configure()</code>	117
8.16.3.6	<code>create()</code>	118
8.16.3.7	<code>getvalue()</code>	118
8.16.3.8	<code>setvalue()</code>	118
8.17	LCARS Namespace Reference	119
8.17.1	Detailed Description	119
8.17.2	Package provided	119
8.18	lcc Namespace Reference	119
8.18.1	Detailed Description	121
8.18.2	Package provided	122
8.18.3	Package provided	122
8.18.4	Package provided	122
8.18.5	Package provided	122
8.18.6	Typedef Documentation	122
8.18.6.1	<code>byte</code>	122
8.18.6.2	<code>bytelist</code>	122
8.18.6.3	<code>bytelist72</code>	122
8.18.6.4	<code>databuf</code>	123
8.18.6.5	<code>eightbytes</code>	123
8.18.6.6	<code>fifteenbits</code>	123
8.18.6.7	<code>fivebits</code>	123
8.18.6.8	<code>headerword</code>	123
8.18.6.9	<code>length</code>	123
8.18.6.10	<code>nid</code>	124

8.18.6.11 sixbits . . . . .	124
8.18.6.12 sixteenbits . . . . .	124
8.18.6.13 threebits . . . . .	124
8.18.6.14 twelvebits . . . . .	124
8.18.6.15 twobits . . . . .	124
8.18.6.16 uint32 . . . . .	125
8.18.7 Enumeration Type Documentation . . . . .	125
8.18.7.1 datagramcontent . . . . .	125
8.18.7.2 eventvalidity . . . . .	125
8.18.8 Function Documentation . . . . .	125
8.18.8.1 AbstractMessage() . . . . .	126
8.18.8.2 AbstractMRMessage() . . . . .	126
8.19 linuxgpio Namespace Reference . . . . .	126
8.19.1 Detailed Description . . . . .	127
8.19.2 Package provided . . . . .	127
8.19.3 Typedef Documentation . . . . .	127
8.19.3.1 pinnotype . . . . .	127
8.19.4 Enumeration Type Documentation . . . . .	127
8.19.4.1 pindirection . . . . .	127
8.20 nce Namespace Reference . . . . .	128
8.20.1 Detailed Description . . . . .	129
8.20.2 Package provided . . . . .	129
8.20.3 Typedef Documentation . . . . .	129
8.20.3.1 AccessoryNumber . . . . .	130
8.20.3.2 AspectBits . . . . .	130
8.20.3.3 CabNumber . . . . .	130
8.20.3.4 ConsistAddress . . . . .	130
8.20.3.5 CSAddress . . . . .	130

8.20.3.6	EchoMode	131
8.20.3.7	Hours	131
8.20.3.8	LCDMessage16	131
8.20.3.9	LCDMessage8	131
8.20.3.10	LocoAddress	131
8.20.3.11	MacroNumber	132
8.20.3.12	Minutes	132
8.20.3.13	MomentumLevel	132
8.20.3.14	RAMData	132
8.20.3.15	RAMData8	132
8.20.3.16	RawPacket	132
8.20.3.17	RawTrackPacket	133
8.20.3.18	ScaleClockRatio	133
8.20.3.19	Speed128	133
8.20.3.20	Speed28	133
8.20.3.21	UByte	133
8.20.4	Enumeration Type Documentation	133
8.20.4.1	Direction	133
8.20.4.2	SpeedMode	134
8.20.5	Function Documentation	134
8.20.5.1	ErrorMessage()	134
8.21	OvalWidgets Namespace Reference	135
8.21.1	Detailed Description	136
8.21.2	Package provided	136
8.21.3	Function Documentation	136
8.21.3.1	_ConfigureFont()	136
8.21.3.2	_ConfigureText()	136
8.21.3.3	_UnderSplit()	137



8.21.3.4	<a href="#">_VerifyFont()</a>	137
8.21.3.5	<a href="#">_VerifyIntegerOrEmpty()</a>	138
8.21.3.6	<a href="#">ColorFillOption()</a>	138
8.21.3.7	<a href="#">ColorOptionMethods()</a>	138
8.21.3.8	<a href="#">ColorOutlineOption()</a>	138
8.21.3.9	<a href="#">CommonValidateMethods()</a>	139
8.21.3.10	<a href="#">FontFamily()</a>	139
8.21.3.11	<a href="#">OvalLabel()</a>	139
8.21.3.12	<a href="#">SquareEndOptions()</a>	140
8.21.3.13	<a href="#">XYWH()</a>	140
8.21.3.14	<a href="#">~OvalLabel()</a>	140
8.21.4	<a href="#">Variable Documentation</a>	140
8.21.4.1	<a href="#">canvas</a>	140
8.21.4.2	<a href="#">HBar</a>	141
8.21.4.3	<a href="#">VBar</a>	141
8.22	<a href="#">PanedWindow Namespace Reference</a>	141
8.22.1	<a href="#">Detailed Description</a>	141
8.22.2	<a href="#">Package provided</a>	141
8.23	<a href="#">Parsers Namespace Reference</a>	141
8.23.1	<a href="#">Detailed Description</a>	142
8.23.2	<a href="#">Tcl Package Provided</a>	143
8.23.3	<a href="#">Library Provided</a>	143
8.24	<a href="#">raildriver Namespace Reference</a>	143
8.24.1	<a href="#">Detailed Description</a>	144
8.24.2	<a href="#">Package provided</a>	144
8.24.3	<a href="#">Typedef Documentation</a>	144
8.24.3.1	<a href="#">eventlist</a>	144
8.24.4	<a href="#">Enumeration Type Documentation</a>	144

8.24.4.1 RaildriverEvents . . . . .	144
8.25 ReadConfiguration Namespace Reference . . . . .	145
8.25.1 Detailed Description . . . . .	145
8.25.2 Package provided . . . . .	145
8.25.3 Function Documentation . . . . .	146
8.25.3.1 ConfigurationType() . . . . .	146
8.25.3.2 IsEven() . . . . .	146
8.25.3.3 ReadConfiguration() . . . . .	147
8.25.3.4 WriteConfiguration() . . . . .	147
8.26 TTSupport Namespace Reference . . . . .	148
8.26.1 Detailed Description . . . . .	149
8.26.2 Tcl Package Provided . . . . .	149
8.26.3 Library Provided . . . . .	149
8.26.4 Typedef Documentation . . . . .	149
8.26.4.1 doubleVector . . . . .	150
8.26.4.2 OptionHashMap . . . . .	150
8.26.4.3 StringList . . . . .	150
8.26.4.4 stringVector . . . . .	150
8.26.4.5 TrainList . . . . .	151
8.26.4.6 TrainStationTimes . . . . .	151
8.26.4.7 TrainTimesAtStation . . . . .	151
8.26.5 Function Documentation . . . . .	151
8.26.5.1 StringListFromString() . . . . .	151
8.26.5.2 StringListToString() . . . . .	152
8.27 xpressnet Namespace Reference . . . . .	152
8.27.1 Detailed Description . . . . .	154
8.27.2 Package provided . . . . .	154
8.27.3 Typedef Documentation . . . . .	155

8.27.3.1	ConsistAddress	155
8.27.3.2	DecoderLongAddress	155
8.27.3.3	ElementAddress	155
8.27.3.4	nibble	155
8.27.3.5	S_128	155
8.27.3.6	S_14	156
8.27.3.7	S_27	156
8.27.3.8	S_28	156
8.27.3.9	u10	156
8.27.3.10	u3	156
8.27.3.11	u7	156
8.27.3.12	ubyte	157
8.27.4	Enumeration Type Documentation	157
8.27.4.1	DirectionCode	157
8.27.4.2	ErrorTypeCode	157
8.27.4.3	MessageTypeCode	158
8.27.4.4	NibbleCode	158
8.27.4.5	PowerUpMode	158
8.27.4.6	SpeedStepModeCode	159
8.27.4.7	TypeCode	159

<b>9</b>	<b>Class Documentation</b>	<b>161</b>
9.1	xpressnet::AccessoryDecoderInformation Class Reference	161
9.1.1	Detailed Description	162
9.1.2	Constructor & Destructor Documentation	162
9.1.2.1	AccessoryDecoderInformation()	162
9.1.3	Member Function Documentation	162
9.1.3.1	AccessoryType()	162
9.1.3.2	Address()	164
9.1.3.3	Completed()	164
9.1.3.4	Nibble()	164
9.1.3.5	NumberOfFeedbackElements()	165
9.1.3.6	TurnoutStatus()	165
9.1.4	Member Data Documentation	165
9.1.4.1	_accessory_type	165
9.1.4.2	_address	165
9.1.4.3	_completed	166
9.1.4.4	_nibble	166
9.1.4.5	_numberOfFeedbackElements	166
9.1.4.6	_t1	166
9.1.4.7	_t2	166
9.2	Instruments::AnalogClock Class Reference	166
9.2.1	Detailed Description	166
9.2.2	Constructor & Destructor Documentation	167
9.2.2.1	AnalogClock()	167
9.2.2.2	~AnalogClock()	168
9.2.3	Member Function Documentation	168
9.2.3.1	settime()	168
9.3	azatrax::Azatrax Class Reference	168

9.3.1	Detailed Description	171
9.3.2	Member Enumeration Documentation	171
9.3.2.1	anonymous enum	171
9.3.2.2	commands	171
9.3.2.3	DeviceConnectionList	174
9.3.3	Constructor & Destructor Documentation	174
9.3.3.1	Azatrax()	174
9.3.3.2	~Azatrax()	174
9.3.4	Member Function Documentation	174
9.3.4.1	AllConnectedDevices()	175
9.3.4.2	GetProductId()	175
9.3.4.3	GetStateData()	175
9.3.4.4	Identify_1()	176
9.3.4.5	IsThisTheAzatraxWeAreLookingFor()	176
9.3.4.6	MyProduct()	176
9.3.4.7	MyProductId()	177
9.3.4.8	NumberOfOpenDevices()	177
9.3.4.9	OpenDevice()	177
9.3.4.10	PacketCount()	177
9.3.4.11	ProductIdCode()	178
9.3.4.12	RestoreLEDFunction()	178
9.3.4.13	send2Bytes()	178
9.3.4.14	send3Bytes()	179
9.3.4.15	sendByte()	179
9.3.4.16	SerialNumber()	180
9.3.5	Friends And Related Function Documentation	180
9.3.5.1	MRD	180
9.3.5.2	SL2	180

9.3.5.3	SR4	180
9.3.6	Member Data Documentation	180
9.3.6.1	deviceOpenCount	181
9.3.6.2	handle	181
9.3.6.3	myProductId	181
9.3.6.4	mySerialNumber	181
9.3.6.5	stateDataPacket	182
9.4	Parsers::BezierBody Class Reference	182
9.4.1	Detailed Description	183
9.4.2	Constructor & Destructor Documentation	183
9.4.2.1	BezierBody()	183
9.4.3	Member Function Documentation	183
9.4.3.1	BezierEnds()	183
9.4.3.2	BezierSegmentCount()	184
9.4.3.3	CleanUpBezierBody()	184
9.4.3.4	CleanUpElement()	184
9.4.3.5	ConsBezierBody()	184
9.4.3.6	Element()	184
9.4.4	Friends And Related Function Documentation	185
9.4.4.1	BezierBodyElt	185
9.4.4.2	TrackGraph	185
9.4.5	Member Data Documentation	185
9.4.5.1	element	185
9.4.5.2	next	185
9.5	Parsers::BezierBodyElt Class Reference	185
9.5.1	Detailed Description	187
9.5.2	Member Enumeration Documentation	187
9.5.2.1	BezierBodyEltType	187

9.5.3	Constructor & Destructor Documentation . . . . .	188
9.5.3.1	BezierBodyElt() . . . . .	188
9.5.3.2	~BezierBodyElt() . . . . .	188
9.5.4	Member Function Documentation . . . . .	188
9.5.4.1	GetCurveSegment() . . . . .	188
9.5.4.2	GetStraightSegment() . . . . .	188
9.5.4.3	InitTSegId() . . . . .	189
9.5.4.4	MakeCurveSegment() . . . . .	189
9.5.4.5	MakeStraightSegment() . . . . .	189
9.5.4.6	MakeTrackEnd() . . . . .	189
9.5.4.7	TheType() . . . . .	190
9.5.5	Friends And Related Function Documentation . . . . .	190
9.5.5.1	BezierBody . . . . .	190
9.5.5.2	TrackGraph . . . . .	190
9.5.6	Member Data Documentation . . . . .	190
9.5.6.1	ang0 . . . . .	190
9.5.6.2	ang1 . . . . .	190
9.5.6.3	pos1 . . . . .	191
9.5.6.4	pos2 . . . . .	191
9.5.6.5	radius . . . . .	191
9.5.6.6	segCount . . . . .	191
9.5.6.7	segmentId . . . . .	191
9.5.6.8	theEnd . . . . .	192
9.5.6.9	theType . . . . .	192
9.6	TTSupport::Cab Class Reference . . . . .	192
9.6.1	Detailed Description . . . . .	193
9.6.2	Constructor & Destructor Documentation . . . . .	193
9.6.2.1	Cab() [1/2] . . . . .	193

9.6.2.2	<code>~Cab()</code>	193
9.6.2.3	<code>Cab()</code> [2/2]	194
9.6.3	Member Function Documentation	194
9.6.3.1	<code>Color()</code>	194
9.6.3.2	<code>Name()</code>	194
9.6.3.3	<code>operator=()</code>	194
9.6.3.4	<code>Read()</code>	195
9.6.3.5	<code>Write()</code>	195
9.6.4	Member Data Documentation	195
9.6.4.1	<code>color</code>	195
9.6.4.2	<code>name</code>	196
9.7	Instruments::CabSignalLamp Class Reference	196
9.7.1	Detailed Description	196
9.7.2	Constructor & Destructor Documentation	197
9.7.2.1	<code>CabSignalLamp()</code>	197
9.7.2.2	<code>~CabSignalLamp()</code>	197
9.7.3	Member Function Documentation	198
9.7.3.1	<code>_ConfigureFillColor()</code>	198
9.7.3.2	<code>_ConfigureOutlineColor()</code>	198
9.7.3.3	<code>_ConfigureSize()</code>	198
9.7.3.4	<code>_ConfigureXY()</code>	199
9.7.4	Member Data Documentation	199
9.7.4.1	<code>canvas</code>	199
9.7.4.2	<code>sx</code>	199
9.7.4.3	<code>sy</code>	199
9.8	Icc::CanAlias Class Reference	200
9.8.1	Detailed Description	200
9.8.2	Constructor & Destructor Documentation	201



9.8.2.1	CanAlias()	201
9.8.3	Member Function Documentation	201
9.8.3.1	_peelnid()	201
9.8.3.2	getMyAlias()	201
9.8.3.3	getMyNIDList()	202
9.8.3.4	getNextAlias()	202
9.8.3.5	validate()	202
9.8.4	Member Data Documentation	202
9.8.4.1	lfsr1	202
9.8.4.2	lfsr2	203
9.8.4.3	myalias	203
9.8.4.4	nidlist	203
9.9	Icc::CANGridConnect Class Reference	203
9.9.1	Detailed Description	205
9.9.2	Constructor & Destructor Documentation	205
9.9.2.1	CANGridConnect()	205
9.9.3	Member Function Documentation	206
9.9.3.1	_flags0()	206
9.9.3.2	_messageReader()	206
9.9.3.3	_reserveMyAlias()	207
9.9.3.4	_sendDatagram()	207
9.9.3.5	_sendmessage()	207
9.9.3.6	_timedout()	207
9.9.3.7	countNUL()	208
9.9.3.8	getAliasOfNID()	208
9.9.3.9	getAllAliases()	208
9.9.3.10	getAllNIDs()	209
9.9.3.11	getBits()	209

9.9.3.12	<a href="#">getNIDofAlias()</a>	209
9.9.3.13	<a href="#">listeq()</a> [1/2]	210
9.9.3.14	<a href="#">listeq()</a> [2/2]	210
9.9.3.15	<a href="#">populateAliasMap()</a>	211
9.9.3.16	<a href="#">reserveAlias()</a>	211
9.9.3.17	<a href="#">sendMessage()</a>	211
9.9.3.18	<a href="#">sendOpenLCBMessage()</a>	211
9.9.3.19	<a href="#">setMessageHandler()</a>	212
9.9.3.20	<a href="#">updateAliasMap()</a>	212
9.9.4	<a href="#">Member Data Documentation</a>	212
9.9.4.1	<a href="#">_timeout</a>	213
9.9.4.2	<a href="#">_timeoutFlag</a>	213
9.9.4.3	<a href="#">aliasMap</a>	213
9.9.4.4	<a href="#">canheader</a>	213
9.9.4.5	<a href="#">datagrambuffers</a>	213
9.9.4.6	<a href="#">gcmessage</a>	213
9.9.4.7	<a href="#">gcreply</a>	214
9.9.4.8	<a href="#">messagebuffers</a>	214
9.9.4.9	<a href="#">messagehandler</a>	214
9.9.4.10	<a href="#">mtidetail</a>	214
9.9.4.11	<a href="#">mtiheader</a>	214
9.9.4.12	<a href="#">mycanalias</a>	214
9.9.4.13	<a href="#">nidMap</a>	215
9.9.4.14	<a href="#">NIDPATTERN</a>	215
9.9.4.15	<a href="#">parent</a>	215
9.9.4.16	<a href="#">simplenodeflags</a>	215
9.10	<a href="#">Icc::CANGridConnectOverCANSocket Class Reference</a>	215
9.10.1	<a href="#">Detailed Description</a>	216

9.10.2	Constructor & Destructor Documentation	217
9.10.2.1	CANGridConnectOverCANSocket()	217
9.10.3	Member Function Documentation	217
9.10.3.1	_CancelOpenTransport()	217
9.10.3.2	_OpenTransport()	218
9.10.3.3	buildSocketnamenidDialog()	218
9.10.3.4	drawOptionsDialog()	218
9.10.3.5	requiredOpts()	219
9.10.4	Member Data Documentation	219
9.10.4.1	gccomponent	219
9.10.4.2	nidLEntry	219
9.10.4.3	socket	219
9.10.4.4	socketnameLEntry	220
9.10.4.5	socketnamenidDialog	220
9.11	Icc::CANGridConnectOverTcp Class Reference	220
9.11.1	Detailed Description	221
9.11.2	Constructor & Destructor Documentation	221
9.11.2.1	CANGridConnectOverTcp()	221
9.11.3	Member Function Documentation	222
9.11.3.1	_CancelOpenTransport()	222
9.11.3.2	_OpenTransport()	222
9.11.3.3	buildPortnidandhostDialog()	222
9.11.3.4	drawOptionsDialog()	222
9.11.3.5	requiredOpts()	223
9.11.4	Member Data Documentation	223
9.11.4.1	gccomponent	223
9.11.4.2	hostLEntry	224
9.11.4.3	nidLEntry	224

9.11.4.4	portLSpin . . . . .	224
9.11.4.5	portnidandhostDialog . . . . .	224
9.11.4.6	socket . . . . .	224
9.12	lcc::CANGridConnectOverUSBSerial Class Reference . . . . .	224
9.12.1	Detailed Description . . . . .	225
9.12.2	Constructor & Destructor Documentation . . . . .	226
9.12.2.1	CANGridConnectOverUSBSerial() . . . . .	226
9.12.3	Member Function Documentation . . . . .	226
9.12.3.1	_CancelOpenTransport() . . . . .	226
9.12.3.2	_OpenTransport() . . . . .	227
9.12.3.3	buildPortandnidDialog() . . . . .	227
9.12.3.4	drawOptionsDialog() . . . . .	227
9.12.3.5	findAvailableComPorts() . . . . .	228
9.12.3.6	requiredOpts() . . . . .	228
9.12.4	Member Data Documentation . . . . .	228
9.12.4.1	gccomponent . . . . .	228
9.12.4.2	nidLEntry . . . . .	229
9.12.4.3	portandnidDialog . . . . .	229
9.12.4.4	portLCombo . . . . .	229
9.12.4.5	ttyfd . . . . .	229
9.13	lcc::CANHeader Class Reference . . . . .	229
9.13.1	Detailed Description . . . . .	230
9.13.2	Constructor & Destructor Documentation . . . . .	230
9.13.2.1	CANHeader() . . . . .	230
9.13.3	Member Function Documentation . . . . .	231
9.13.3.1	getHeader() . . . . .	231
9.13.3.2	setHeader() . . . . .	231
9.13.4	Member Data Documentation . . . . .	231

9.13.4.1	OPENLCBFRAME_MASK . . . . .	231
9.13.4.2	OPENLCBFRAME_SHIFT . . . . .	232
9.13.4.3	RESERVED_SHIFT . . . . .	232
9.13.4.4	SRCID_MASK . . . . .	232
9.13.4.5	SRCID_SHIFT . . . . .	232
9.13.4.6	VARIABLEFIELD_MASK . . . . .	232
9.13.4.7	VARIABLEFIELD_SHIFT . . . . .	232
9.14	Icc::CanMessage Class Reference . . . . .	233
9.14.1	Detailed Description . . . . .	234
9.14.2	Constructor & Destructor Documentation . . . . .	235
9.14.2.1	CanMessage() . . . . .	235
9.14.3	Member Function Documentation . . . . .	236
9.14.3.1	copy() . . . . .	236
9.14.3.2	equals() . . . . .	236
9.14.3.3	getData() . . . . .	236
9.14.3.4	getHeader() . . . . .	237
9.14.3.5	hashCode() . . . . .	237
9.14.3.6	replyExpected() . . . . .	237
9.14.3.7	setData() . . . . .	237
9.14.3.8	setHeader() . . . . .	238
9.14.3.9	setNumDataElements() . . . . .	238
9.14.3.10	toString() . . . . .	238
9.14.3.11	validate() . . . . .	239
9.14.4	Member Data Documentation . . . . .	239
9.14.4.1	_header . . . . .	239
9.15	Icc::CanTransport Class Reference . . . . .	239
9.15.1	Detailed Description . . . . .	240
9.15.2	Constructor & Destructor Documentation . . . . .	240

9.15.2.1	CanTransport()	240
9.15.3	Member Function Documentation	240
9.15.3.1	_transportlayerconf()	241
9.15.4	Member Data Documentation	241
9.15.4.1	transport	241
9.16	FCFSupport::Car Class Reference	241
9.16.1	Detailed Description	245
9.16.2	Constructor & Destructor Documentation	245
9.16.2.1	Car() [1/3]	245
9.16.2.2	Car() [2/3]	245
9.16.2.3	Car() [3/3]	246
9.16.3	Member Function Documentation	247
9.16.3.1	Assignments()	247
9.16.3.2	CarOwner()	247
9.16.3.3	ClearAssignments()	247
9.16.3.4	ClearMovementsThisSession()	248
9.16.3.5	ClearTrips()	248
9.16.3.6	Destination()	248
9.16.3.7	Divisions()	248
9.16.3.8	EmptyP()	248
9.16.3.9	FixedRouteP()	249
9.16.3.10	IncrementAssignments()	249
9.16.3.11	IncrementTrips()	249
9.16.3.12	IncrmentMovementsThisSession()	249
9.16.3.13	IsDoneP()	249
9.16.3.14	LastTrain()	250
9.16.3.15	LdLmt()	250
9.16.3.16	Length()	250

9.16.3.17 Load()	250
9.16.3.18 LoadedP()	250
9.16.3.19 Location()	251
9.16.3.20 LtWt()	251
9.16.3.21 Marks()	251
9.16.3.22 MovementsThisSession()	251
9.16.3.23 Number()	251
9.16.3.24 OkToMirrorP()	252
9.16.3.25 operator=()	252
9.16.3.26 Peek()	252
9.16.3.27 Plate()	252
9.16.3.28 PrevTrain()	253
9.16.3.29 SetAssignments()	253
9.16.3.30 SetCarOwner()	253
9.16.3.31 SetDestination()	253
9.16.3.32 SetDivisions()	254
9.16.3.33 SetDone()	254
9.16.3.34 SetFixedRouteP()	254
9.16.3.35 SetLastTrain()	254
9.16.3.36 SetLdLmt()	255
9.16.3.37 SetLength()	255
9.16.3.38 SetLocation()	255
9.16.3.39 SetLtWt()	255
9.16.3.40 SetMarks()	256
9.16.3.41 SetNotDone()	256
9.16.3.42 SetNumber()	256
9.16.3.43 SetOkToMirrorP()	256
9.16.3.44 SetPeek()	257

9.16.3.45 SetPlate()	257
9.16.3.46 SetPrevTrain()	257
9.16.3.47 SetType()	257
9.16.3.48 SetWeightClass()	258
9.16.3.49 Trips()	258
9.16.3.50 Type()	258
9.16.3.51 UnLoad()	258
9.16.3.52 WeightClass()	258
9.16.4 Friends And Related Function Documentation	259
9.16.4.1 System	259
9.16.5 Member Data Documentation	259
9.16.5.1 assignments	259
9.16.5.2 destination	259
9.16.5.3 divisions	259
9.16.5.4 doneP	260
9.16.5.5 fixedP	260
9.16.5.6 lasttrain	260
9.16.5.7 ldlmt	260
9.16.5.8 length	260
9.16.5.9 loadedP	261
9.16.5.10 location	261
9.16.5.11 ltwt	261
9.16.5.12 marks	261
9.16.5.13 mirrorP	261
9.16.5.14 moves	262
9.16.5.15 number	262
9.16.5.16 owner	262
9.16.5.17 peek	262



9.16.5.18 plate	262
9.16.5.19 prevtrain	263
9.16.5.20 tmpStatus	263
9.16.5.21 trips	263
9.16.5.22 type	263
9.16.5.23 weightclass	263
9.17 FCFSupport::CarGroup Class Reference	264
9.17.1 Detailed Description	264
9.17.2 Member Enumeration Documentation	264
9.17.2.1 CarGroupConsts	264
9.17.3 Constructor & Destructor Documentation	265
9.17.3.1 CarGroup() [1/3]	265
9.17.3.2 CarGroup() [2/3]	265
9.17.3.3 CarGroup() [3/3]	265
9.17.4 Member Function Documentation	266
9.17.4.1 Description()	266
9.17.4.2 Group()	266
9.17.4.3 operator=()	266
9.17.5 Member Data Documentation	267
9.17.5.1 description	267
9.17.5.2 group	267
9.18 FCFSupport::CarType Class Reference	267
9.18.1 Detailed Description	268
9.18.2 Member Enumeration Documentation	268
9.18.2.1 CarTypeConsts	268
9.18.3 Constructor & Destructor Documentation	269
9.18.3.1 CarType() [1/3]	269
9.18.3.2 CarType() [2/3]	269

9.18.3.3	CarType() [3/3]	269
9.18.3.4	~CarType()	270
9.18.4	Member Function Documentation	270
9.18.4.1	Comment()	270
9.18.4.2	Group()	270
9.18.4.3	operator=()	270
9.18.4.4	Type()	271
9.18.5	Friends And Related Function Documentation	271
9.18.5.1	System	271
9.18.6	Member Data Documentation	271
9.18.6.1	comment	271
9.18.6.2	group	272
9.18.6.3	type	272
9.19	FCFSupport::PDFFileStructures::CatalogDictionary Class Reference	272
9.19.1	Detailed Description	273
9.19.2	Constructor & Destructor Documentation	273
9.19.2.1	CatalogDictionary()	273
9.19.2.2	~CatalogDictionary()	273
9.19.3	Member Function Documentation	274
9.19.3.1	AddPage()	274
9.19.3.2	AddPageLabelDictionary()	274
9.19.3.3	AddPageLabelTree()	274
9.19.3.4	AddPageTree()	275
9.19.3.5	WriteDictionaryElements()	275
9.19.4	Member Data Documentation	275
9.19.4.1	labels	275
9.19.4.2	pages	276
9.20	cmri::CMri Class Reference	276

9.20.1 Detailed Description . . . . .	277
9.20.2 Constructor & Destructor Documentation . . . . .	278
9.20.2.1 CMri() . . . . .	278
9.20.2.2 ~CMri() . . . . .	278
9.20.3 Member Function Documentation . . . . .	278
9.20.3.1 _readbyte() . . . . .	279
9.20.3.2 _readevent() . . . . .	279
9.20.3.3 _transmit() . . . . .	279
9.20.3.4 InitBoard() . . . . .	279
9.20.3.5 Inputs() . . . . .	280
9.20.3.6 Outputs() . . . . .	281
9.20.4 Member Data Documentation . . . . .	281
9.20.4.1 _timeout . . . . .	281
9.20.4.2 AddressCode . . . . .	281
9.20.4.3 CardType_Byte . . . . .	281
9.20.4.4 DLE . . . . .	282
9.20.4.5 ETX . . . . .	282
9.20.4.6 Init . . . . .	282
9.20.4.7 Poll . . . . .	282
9.20.4.8 Read . . . . .	282
9.20.4.9 STX . . . . .	282
9.20.4.10 Transmit . . . . .	283
9.20.4.11 ttyfd . . . . .	283
9.21 CmriSupport::CmriNode Class Reference . . . . .	283
9.21.1 Detailed Description . . . . .	284
9.21.2 Constructor & Destructor Documentation . . . . .	284
9.21.2.1 CmriNode() . . . . .	284
9.21.3 Member Function Documentation . . . . .	285

9.21.3.1	<code>_ValidateAddress()</code>	285
9.21.3.2	<code>_ValidateByte()</code>	285
9.21.3.3	<code>_ValidateListOfBytes()</code>	287
9.21.3.4	<code>_ValidateSixElementListOfBytes()</code>	287
9.21.3.5	<code>_ValidateType()</code>	287
9.21.3.6	<code>_ValidateWord()</code>	288
9.21.3.7	<code>closeport()</code>	288
9.21.3.8	<code>inputs()</code>	288
9.21.3.9	<code>openport()</code>	288
9.21.3.10	<code>outputs()</code>	289
9.21.3.11	<code>portopenp()</code>	289
9.21.3.12	<code>setbitfield()</code>	289
9.21.3.13	<code>setport()</code>	290
9.21.3.14	<code>validate()</code>	290
9.21.4	Member Data Documentation	290
9.21.4.1	<code>_TypeCodes</code>	290
9.21.4.2	<code>outputbuffer</code>	290
9.22	CTCPanel::CodeButton Class Reference	291
9.22.1	Detailed Description	291
9.22.2	Constructor & Destructor Documentation	292
9.22.2.1	<code>CodeButton()</code>	292
9.22.2.2	<code>~CodeButton()</code>	292
9.22.3	Member Function Documentation	292
9.22.3.1	<code>geti()</code>	292
9.22.3.2	<code>getv()</code>	293
9.22.3.3	<code>invoke()</code>	293
9.22.3.4	<code>seti()</code>	293
9.22.3.5	<code>setv()</code>	293

9.22.4	Member Data Documentation	293
9.22.4.1	canvas	293
9.22.4.2	ctcpanel	294
9.23	xpressnet::CommandStationResponse Class Reference	294
9.23.1	Detailed Description	294
9.23.2	Constructor & Destructor Documentation	295
9.23.2.1	CommandStationResponse()	295
9.23.3	Member Function Documentation	295
9.23.3.1	ResponseType()	295
9.23.3.2	TimeStamp()	295
9.23.4	Member Data Documentation	295
9.23.4.1	_time_stamp	296
9.24	xpressnet::CommandStationStatus Class Reference	296
9.24.1	Detailed Description	297
9.24.2	Constructor & Destructor Documentation	297
9.24.2.1	CommandStationStatus()	297
9.24.3	Member Function Documentation	297
9.24.3.1	EmergencyOff()	297
9.24.3.2	EmergencyStop()	297
9.24.3.3	PoweringUp()	298
9.24.3.4	RAMCheckError()	298
9.24.3.5	ServiceMode()	298
9.24.3.6	StartMode()	298
9.24.4	Member Data Documentation	298
9.24.4.1	_emergency_off	298
9.24.4.2	_emergency_stop	299
9.24.4.3	_poweringup	299
9.24.4.4	_RAM_check_error	299

9.24.4.5	<code>_service_mode</code>	299
9.24.4.6	<code>_start_mode</code>	299
9.25	Parsers::TrackGraph::CompressedEdgeValues Struct Reference	299
9.25.1	Detailed Description	300
9.25.2	Constructor & Destructor Documentation	300
9.25.2.1	<code>CompressedEdgeValues()</code>	300
9.25.3	Member Data Documentation	300
9.25.3.1	<code>length</code>	300
9.26	Parsers::TrackGraph::CompressedNodeValues Struct Reference	300
9.26.1	Detailed Description	301
9.26.2	Constructor & Destructor Documentation	301
9.26.2.1	<code>CompressedNodeValues()</code>	301
9.26.3	Member Function Documentation	301
9.26.3.1	<code>FindSegmentIndex()</code>	302
9.26.4	Member Data Documentation	302
9.26.4.1	<code>id</code>	302
9.26.4.2	<code>position</code>	302
9.26.4.3	<code>rawnode</code>	302
9.26.4.4	<code>segments</code>	302
9.27	Icc::ConfigMemory Class Reference	303
9.27.1	Detailed Description	304
9.27.2	Constructor & Destructor Documentation	305
9.27.2.1	<code>ConfigMemory()</code>	305
9.27.3	Member Function Documentation	305
9.27.3.1	<code>_Close()</code>	305
9.27.3.2	<code>_datagramhandler()</code>	305
9.27.3.3	<code>_Dump()</code>	306
9.27.3.4	<code>_dumpAsHex()</code>	306

9.27.3.5	<a href="#">_dumpAsText()</a>	306
9.27.3.6	<a href="#">_getAddressRange()</a>	307
9.27.3.7	<a href="#">_messagehandler()</a>	307
9.27.3.8	<a href="#">_Read()</a>	307
9.27.3.9	<a href="#">_readmemory()</a>	308
9.27.3.10	<a href="#">_Restore()</a>	308
9.27.3.11	<a href="#">_Write()</a>	308
9.27.3.12	<a href="#">_writememory()</a>	308
9.27.3.13	<a href="#">putdebug()</a>	309
9.27.4	<a href="#">Member Data Documentation</a>	309
9.27.4.1	<a href="#">_datagramrejecterror</a>	309
9.27.4.2	<a href="#">_ioComplete</a>	309
9.27.4.3	<a href="#">_spaces</a>	310
9.27.4.4	<a href="#">address</a>	310
9.27.4.5	<a href="#">count</a>	310
9.27.4.6	<a href="#">datagrambuffer</a>	310
9.27.4.7	<a href="#">olddatagramhandler</a>	310
9.27.4.8	<a href="#">oldgeneralmessagehandler</a>	310
9.27.4.9	<a href="#">readlist</a>	311
9.27.4.10	<a href="#">space</a>	311
9.27.4.11	<a href="#">writelst</a>	311
9.27.4.12	<a href="#">writeReplyCheck</a>	311
9.28	<a href="#">Icc::ConfigOptions Class Reference</a>	311
9.28.1	<a href="#">Detailed Description</a>	312
9.28.2	<a href="#">Constructor &amp; Destructor Documentation</a>	312
9.28.2.1	<a href="#">ConfigOptions()</a>	312
9.28.3	<a href="#">Member Function Documentation</a>	313
9.28.3.1	<a href="#">_Close()</a>	313

9.28.3.2	putdebug()	313
9.28.4	Member Data Documentation	314
9.28.4.1	available	314
9.28.4.2	highest	314
9.28.4.3	lowest	314
9.28.4.4	name	314
9.28.4.5	nodeid	314
9.28.4.6	writelengths	315
9.29	Icc::ConfigurationEditor Class Reference	315
9.29.1	Detailed Description	318
9.29.2	Constructor & Destructor Documentation	318
9.29.2.1	ConfigurationEditor()	318
9.29.3	Member Function Documentation	319
9.29.3.1	_close()	319
9.29.3.2	_datagramhandler()	319
9.29.3.3	_eventidComboRead()	320
9.29.3.4	_eventidComboWrite()	320
9.29.3.5	_eventidEntryRead()	321
9.29.3.6	_eventidEntryWrite()	321
9.29.3.7	_intComboRead()	321
9.29.3.8	_intComboWrite()	322
9.29.3.9	_intSpinRead()	322
9.29.3.10	_intSpinWrite()	323
9.29.3.11	_printexport()	323
9.29.3.12	_printexport_csv()	324
9.29.3.13	_printexport_csv_frame()	324
9.29.3.14	_printexport_csv_frameAcross()	324
9.29.3.15	_printexport_csv_framesAcross()	325



9.29.3.16 _printexport_csv_vframe()	325
9.29.3.17 _printexport_csv_vframeAcross()	326
9.29.3.18 _printexport_pdf()	326
9.29.3.19 _printexport_pdf_frame()	326
9.29.3.20 _printexport_pdf_newpage()	327
9.29.3.21 _printexport_pdf_vframe()	327
9.29.3.22 _printexport_txt()	328
9.29.3.23 _printexport_txt_frame()	328
9.29.3.24 _printexport_txt_vframe()	329
9.29.3.25 _printexport_xml()	329
9.29.3.26 _printexport_xml_frame()	329
9.29.3.27 _printexport_xml_vframe()	331
9.29.3.28 _processXMLnode()	331
9.29.3.29 _readall()	332
9.29.3.30 _readmemory()	332
9.29.3.31 _stringComboRead()	333
9.29.3.32 _stringComboWrite()	333
9.29.3.33 _stringEntryRead()	333
9.29.3.34 _stringEntryWrite()	334
9.29.3.35 _writememory()	334
9.29.3.36 putdebug()	335
9.29.4 Member Data Documentation	335
9.29.4.1 _datagramrejecterror	335
9.29.4.2 _eventidnumber	335
9.29.4.3 _groupnumber	335
9.29.4.4 _intnumber	336
9.29.4.5 _ioComplete	336
9.29.4.6 _menu	336

9.29.4.7	<a href="#">_mkbuttons</a>	336
9.29.4.8	<a href="#">_readall</a>	336
9.29.4.9	<a href="#">_segmentnumber</a>	336
9.29.4.10	<a href="#">_stringnumber</a>	337
9.29.4.11	<a href="#">buttons</a>	337
9.29.4.12	<a href="#">cdi</a>	337
9.29.4.13	<a href="#">datagrambuffer</a>	337
9.29.4.14	<a href="#">editframe</a>	337
9.29.4.15	<a href="#">idheaders</a>	337
9.29.4.16	<a href="#">main</a>	338
9.29.4.17	<a href="#">olddatagramhandler</a>	338
9.29.4.18	<a href="#">printexportfiletypes</a>	338
9.29.4.19	<a href="#">scroll</a>	338
9.29.4.20	<a href="#">statusline</a>	338
9.29.4.21	<a href="#">writeReplyCheck</a>	338
9.30	<a href="#">Parsers::CornuBody Class Reference</a>	339
9.30.1	<a href="#">Detailed Description</a>	340
9.30.2	<a href="#">Constructor &amp; Destructor Documentation</a>	340
9.30.2.1	<a href="#">CornuBody()</a>	340
9.30.3	<a href="#">Member Function Documentation</a>	340
9.30.3.1	<a href="#">CleanUpCornuBody()</a>	340
9.30.3.2	<a href="#">CleanUpElement()</a>	340
9.30.3.3	<a href="#">ConcatCornuBody()</a>	341
9.30.3.4	<a href="#">ConsCornuBody()</a>	341
9.30.3.5	<a href="#">CornuEnds()</a>	341
9.30.3.6	<a href="#">CornuSegmentCount()</a>	341
9.30.3.7	<a href="#">Element()</a>	341
9.30.4	<a href="#">Friends And Related Function Documentation</a>	342

9.30.4.1	CornuBodyElt	342
9.30.4.2	TrackGraph	342
9.30.5	Member Data Documentation	342
9.30.5.1	element	342
9.30.5.2	next	342
9.31	Parsers::CornuBodyElt Class Reference	342
9.31.1	Detailed Description	344
9.31.2	Member Enumeration Documentation	344
9.31.2.1	CornuBodyEltType	344
9.31.3	Constructor & Destructor Documentation	345
9.31.3.1	CornuBodyElt()	345
9.31.3.2	~CornuBodyElt()	345
9.31.4	Member Function Documentation	345
9.31.4.1	GetCurveSegment()	345
9.31.4.2	GetStraightSegment()	345
9.31.4.3	InitTSegId()	346
9.31.4.4	MakeCurveSegment()	346
9.31.4.5	MakeStraightSegment()	346
9.31.4.6	MakeTrackEnd()	346
9.31.4.7	TheType()	347
9.31.5	Friends And Related Function Documentation	347
9.31.5.1	CornuBody	347
9.31.5.2	TrackGraph	347
9.31.6	Member Data Documentation	347
9.31.6.1	ang0	347
9.31.6.2	ang1	347
9.31.6.3	pos1	348
9.31.6.4	pos2	348

9.31.6.5	radius	348
9.31.6.6	segCount	348
9.31.6.7	segmentId	348
9.31.6.8	theEnd	349
9.31.6.9	theType	349
9.32	CTCPanel::Crossing Class Reference	349
9.32.1	Detailed Description	350
9.32.2	Constructor & Destructor Documentation	351
9.32.2.1	Crossing()	351
9.32.2.2	~Crossing()	351
9.32.3	Member Function Documentation	351
9.32.3.1	_configureLabel()	351
9.32.3.2	_VerifyCrossingType()	352
9.32.3.3	geti()	352
9.32.3.4	getv()	352
9.32.3.5	invoke()	352
9.32.3.6	seti()	352
9.32.3.7	setv()	352
9.32.4	Member Data Documentation	353
9.32.4.1	canvas	353
9.32.4.2	ctcpanel	353
9.33	CTCPanel::Crossover Class Reference	353
9.33.1	Detailed Description	354
9.33.2	Constructor & Destructor Documentation	355
9.33.2.1	Crossover()	355
9.33.2.2	~Crossover()	355
9.33.3	Member Function Documentation	355
9.33.3.1	_configureLabel()	356

9.33.3.2	<a href="#">geti()</a>	356
9.33.3.3	<a href="#">getv()</a>	356
9.33.3.4	<a href="#">invoke()</a>	356
9.33.3.5	<a href="#">seti()</a>	356
9.33.3.6	<a href="#">setv()</a>	356
9.33.4	<a href="#">Member Data Documentation</a>	357
9.33.4.1	<a href="#">canvas</a>	357
9.33.4.2	<a href="#">ctcpanel</a>	357
9.33.4.3	<a href="#">state</a>	357
9.34	<a href="#">FCFSupport::PDFFileStructures::CrossReferenceTable Class Reference</a>	357
9.34.1	<a href="#">Detailed Description</a>	358
9.34.2	<a href="#">Member Typedef Documentation</a>	359
9.34.2.1	<a href="#">objectMap</a>	359
9.34.3	<a href="#">Constructor &amp; Destructor Documentation</a>	359
9.34.3.1	<a href="#">CrossReferenceTable()</a>	359
9.34.3.2	<a href="#">~CrossReferenceTable()</a>	359
9.34.4	<a href="#">Member Function Documentation</a>	359
9.34.4.1	<a href="#">AddIndirectObjectToTable()</a>	359
9.34.4.2	<a href="#">FreeObject()</a>	360
9.34.4.3	<a href="#">HighestObjectNumber()</a>	360
9.34.4.4	<a href="#">WriteTable()</a>	360
9.34.5	<a href="#">Friends And Related Function Documentation</a>	361
9.34.5.1	<a href="#">IndirectObject</a>	361
9.34.6	<a href="#">Member Data Documentation</a>	361
9.34.6.1	<a href="#">lastObjectNumber</a>	361
9.34.6.2	<a href="#">objectTable</a>	361
9.35	<a href="#">CTCPanel::CTCLabel Class Reference</a>	361
9.35.1	<a href="#">Detailed Description</a>	362

9.35.2	Constructor & Destructor Documentation	363
9.35.2.1	CTCLabel()	363
9.35.2.2	~CTCLabel()	363
9.35.3	Member Function Documentation	363
9.35.3.1	_configureColor()	364
9.35.3.2	_configureLabel()	364
9.35.3.3	geti()	364
9.35.3.4	getv()	364
9.35.3.5	invoke()	364
9.35.3.6	seti()	365
9.35.3.7	setv()	365
9.35.4	Member Data Documentation	365
9.35.4.1	canvas	365
9.35.4.2	ctcpanel	365
9.36	CTCPanel::CTCPanel Class Reference	365
9.36.1	Detailed Description	369
9.36.2	Constructor & Destructor Documentation	369
9.36.2.1	CTCPanel()	369
9.36.3	Member Function Documentation	370
9.36.3.1	_crosshairEnd()	370
9.36.3.2	_crosshairMove()	370
9.36.3.3	_crosshairStart()	371
9.36.3.4	_CtcMainHScroll2()	371
9.36.3.5	_CtcMainSyncX()	371
9.36.3.6	_PosInteger()	372
9.36.3.7	bind()	372
9.36.3.8	checkInitCP()	373
9.36.3.9	controls_crosshair()	373

9.36.3.10 coords()	373
9.36.3.11 cplist()	374
9.36.3.12 create_CodeButton()	374
9.36.3.13 create_Crossing()	374
9.36.3.14 create_Crossover()	374
9.36.3.15 create_CTCLabel()	375
9.36.3.16 create_CurvedBlock()	375
9.36.3.17 create_DoubleSlip()	375
9.36.3.18 create_EndBumper()	376
9.36.3.19 create_HiddenBlock()	376
9.36.3.20 create_Lamp()	376
9.36.3.21 create_PushButton()	377
9.36.3.22 create_SchLabel()	377
9.36.3.23 create_ScissorCrossover()	377
9.36.3.24 create_Signal()	378
9.36.3.25 create_SIGPlate()	378
9.36.3.26 create_SingleSlip()	378
9.36.3.27 create_StraightBlock()	379
9.36.3.28 create_StubYard()	379
9.36.3.29 create_Switch()	379
9.36.3.30 create_SWPlate()	380
9.36.3.31 create_ThreeWaySW()	380
9.36.3.32 create_ThroughYard()	380
9.36.3.33 create_Toggle()	381
9.36.3.34 delete()	381
9.36.3.35 exists()	381
9.36.3.36 geti()	382
9.36.3.37 getv()	382

9.36.3.38	<code>getZoom()</code>	382
9.36.3.39	<code>invoke()</code>	383
9.36.3.40	<code>itemcget()</code>	383
9.36.3.41	<code>itemconfigure()</code>	383
9.36.3.42	<code>lappendCP()</code>	384
9.36.3.43	<code>lremoveCP()</code>	384
9.36.3.44	<code>move()</code>	384
9.36.3.45	<code>objectlist()</code>	385
9.36.3.46	<code>print()</code>	385
9.36.3.47	<code>schematic_crosshair()</code>	385
9.36.3.48	<code>seti()</code>	386
9.36.3.49	<code>setv()</code>	386
9.36.3.50	<code>setZoom()</code>	387
9.36.3.51	<code>updateAndSyncCP()</code>	387
9.36.3.52	<code>updateSR()</code>	387
9.36.3.53	<code>zoomBy()</code>	388
9.36.4	Member Data Documentation	388
9.36.4.1	<code>_ch_oldfocus</code>	388
9.36.4.2	<code>_ch_oldgrab</code>	388
9.36.4.3	<code>controls</code>	388
9.36.4.4	<code>controlsYscroll</code>	389
9.36.4.5	<code>CPData</code>	389
9.36.4.6	<code>CPList</code>	389
9.36.4.7	<code>Objects</code>	389
9.36.4.8	<code>scale</code>	389
9.36.4.9	<code>schematic</code>	389
9.36.4.10	<code>schematicYscroll</code>	390
9.36.4.11	<code>xscroll</code>	390



9.37	ctiacela::CTIAcela Class Reference . . . . .	390
9.37.1	Detailed Description . . . . .	393
9.38	CTCPanel::CurvedBlock Class Reference . . . . .	394
9.38.1	Detailed Description . . . . .	395
9.38.2	Constructor & Destructor Documentation . . . . .	396
9.38.2.1	CurvedBlock() . . . . .	396
9.38.2.2	~CurvedBlock() . . . . .	396
9.38.3	Member Function Documentation . . . . .	396
9.38.3.1	_configureLabel() . . . . .	396
9.38.3.2	_RadiansToDegrees() . . . . .	397
9.38.3.3	_square() . . . . .	397
9.38.3.4	geti() . . . . .	397
9.38.3.5	invoke() . . . . .	397
9.38.3.6	seti() . . . . .	397
9.38.3.7	setv() . . . . .	397
9.38.4	Member Data Documentation . . . . .	398
9.38.4.1	_PI . . . . .	398
9.38.4.2	canvas . . . . .	398
9.38.4.3	ctcpanel . . . . .	398
9.39	Instruments::DialInstrument Class Reference . . . . .	398
9.39.1	Detailed Description . . . . .	399
9.39.2	Constructor & Destructor Documentation . . . . .	400
9.39.2.1	DialInstrument() . . . . .	401
9.39.2.2	~DialInstrument() . . . . .	401
9.39.3	Member Function Documentation . . . . .	401
9.39.3.1	setvalue() . . . . .	401
9.39.4	Member Data Documentation . . . . .	401
9.39.4.1	dTextX . . . . .	402

9.39.4.2	dTextY	402
9.39.4.3	ValueRange	402
9.40	FCFSupport::PDFFileStructures::Dictionary Class Reference	402
9.40.1	Detailed Description	403
9.40.2	Constructor & Destructor Documentation	403
9.40.2.1	Dictionary()	403
9.40.2.2	~Dictionary()	403
9.40.3	Member Function Documentation	404
9.40.3.1	WriteDictionaryElements()	404
9.40.3.2	WriteDirect()	404
9.41	Instruments::DigitalClock Class Reference	404
9.41.1	Detailed Description	405
9.41.2	Constructor & Destructor Documentation	405
9.41.2.1	DigitalClock()	406
9.41.2.2	~DigitalClock()	406
9.41.3	Member Function Documentation	406
9.41.3.1	settime()	406
9.42	Instruments::DigitalInstrument Class Reference	406
9.42.1	Detailed Description	407
9.42.2	Constructor & Destructor Documentation	408
9.42.2.1	DigitalInstrument()	408
9.42.2.2	~DigitalInstrument()	408
9.42.3	Member Function Documentation	408
9.42.3.1	setvalue()	408
9.43	FCFSupport::Division Class Reference	409
9.43.1	Detailed Description	410
9.43.2	Constructor & Destructor Documentation	410
9.43.2.1	Division() [1/3]	410

9.43.2.2	Division() [2/3]	410
9.43.2.3	Division() [3/3]	411
9.43.2.4	~Division()	411
9.43.3	Member Function Documentation	411
9.43.3.1	AppendStation()	411
9.43.3.2	Area()	412
9.43.3.3	Home()	412
9.43.3.4	Name()	412
9.43.3.5	NumberOfStations()	412
9.43.3.6	operator=()	412
9.43.3.7	Symbol()	413
9.43.3.8	TheStation()	413
9.43.4	Friends And Related Function Documentation	413
9.43.4.1	System	413
9.43.5	Member Data Documentation	413
9.43.5.1	area	414
9.43.5.2	home	414
9.43.5.3	name	414
9.43.5.4	stations	414
9.43.5.5	symbol	414
9.44	xpressnet::DoubleHeaderInformation Class Reference	415
9.44.1	Detailed Description	416
9.44.2	Constructor & Destructor Documentation	416
9.44.2.1	DoubleHeaderInformation()	416
9.44.3	Member Function Documentation	417
9.44.3.1	Address()	417
9.44.3.2	Address2()	418
9.44.3.3	Available()	418

9.44.3.4	<a href="#">Direction()</a>	418
9.44.3.5	<a href="#">Function()</a>	418
9.44.3.6	<a href="#">Speed()</a>	418
9.44.3.7	<a href="#">SpeedStepMode()</a>	419
9.44.4	<a href="#">Member Data Documentation</a>	419
9.44.4.1	<a href="#">_address</a>	419
9.44.4.2	<a href="#">_address2</a>	419
9.44.4.3	<a href="#">_available</a>	419
9.44.4.4	<a href="#">_direction</a>	419
9.44.4.5	<a href="#">_function0</a>	420
9.44.4.6	<a href="#">_function1</a>	420
9.44.4.7	<a href="#">_function10</a>	420
9.44.4.8	<a href="#">_function11</a>	420
9.44.4.9	<a href="#">_function12</a>	420
9.44.4.10	<a href="#">_function2</a>	420
9.44.4.11	<a href="#">_function3</a>	421
9.44.4.12	<a href="#">_function4</a>	421
9.44.4.13	<a href="#">_function5</a>	421
9.44.4.14	<a href="#">_function6</a>	421
9.44.4.15	<a href="#">_function7</a>	421
9.44.4.16	<a href="#">_function8</a>	421
9.44.4.17	<a href="#">_function9</a>	422
9.44.4.18	<a href="#">_speed</a>	422
9.44.4.19	<a href="#">_speedstep</a>	422
9.45	<a href="#">xpressnet::DoubleHeaderMuError Class Reference</a>	422
9.45.1	<a href="#">Detailed Description</a>	422
9.45.2	<a href="#">Constructor &amp; Destructor Documentation</a>	423
9.45.2.1	<a href="#">DoubleHeaderMuError()</a>	423

9.45.3	Member Function Documentation	423
9.45.3.1	Error()	423
9.45.4	Member Data Documentation	423
9.45.4.1	_error	423
9.46	CTCPanel::DoubleSlip Class Reference	423
9.46.1	Detailed Description	424
9.46.2	Constructor & Destructor Documentation	425
9.46.2.1	DoubleSlip()	425
9.46.2.2	~DoubleSlip()	426
9.46.3	Member Function Documentation	426
9.46.3.1	_configureLabel()	426
9.46.3.2	geti()	426
9.46.3.3	getv()	426
9.46.3.4	invoke()	427
9.46.3.5	seti()	427
9.46.3.6	setv()	427
9.46.4	Member Data Documentation	427
9.46.4.1	canvas	427
9.46.4.2	ctcpanel	428
9.46.4.3	state	428
9.47	Parsers::TrackGraph::EdgeValues Struct Reference	428
9.47.1	Detailed Description	428
9.47.2	Constructor & Destructor Documentation	429
9.47.2.1	EdgeValues()	429
9.47.3	Member Data Documentation	429
9.47.3.1	a	429
9.47.3.2	index	429
9.47.3.3	length	429

9.47.3.4	x	430
9.47.3.5	y	430
9.48	CTCPanel::EndBumper Class Reference	430
9.48.1	Detailed Description	430
9.48.2	Constructor & Destructor Documentation	431
9.48.2.1	EndBumper()	431
9.48.2.2	~EndBumper()	432
9.48.3	Member Function Documentation	432
9.48.3.1	_configureLabel()	432
9.48.3.2	geti()	432
9.48.3.3	invoke()	432
9.48.3.4	seti()	433
9.48.3.5	setv()	433
9.48.4	Member Data Documentation	433
9.48.4.1	canvas	433
9.48.4.2	ctcpanel	433
9.49	TTSupport::eqstr Struct Reference	433
9.49.1	Member Function Documentation	434
9.49.1.1	operator>()	434
9.50	Icc::EventID Class Reference	434
9.50.1	Detailed Description	434
9.50.2	Constructor & Destructor Documentation	435
9.50.2.1	EventID()	435
9.50.3	Member Function Documentation	435
9.50.3.1	_getEventID()	435
9.50.3.2	_setEventID()	436
9.50.3.3	validate()	436
9.50.4	Member Data Documentation	437

9.50.4.1	<a href="#">_eventID</a>	437
9.50.4.2	<a href="#">EVENTIDFMT</a>	437
9.51	<a href="#">Icc::EventID_or_null Class Reference</a>	437
9.51.1	<a href="#">Detailed Description</a>	437
9.51.2	<a href="#">Member Function Documentation</a>	437
9.51.2.1	<a href="#">validate()</a>	437
9.52	<a href="#">Icc::EventLog Class Reference</a>	438
9.52.1	<a href="#">Detailed Description</a>	438
9.52.2	<a href="#">Constructor &amp; Destructor Documentation</a>	439
9.52.2.1	<a href="#">EventLog()</a>	439
9.52.3	<a href="#">Member Function Documentation</a>	439
9.52.3.1	<a href="#">_clear()</a>	439
9.52.3.2	<a href="#">_close()</a>	439
9.52.3.3	<a href="#">_sendtheevent()</a>	440
9.52.3.4	<a href="#">eventReceived()</a>	440
9.52.3.5	<a href="#">open()</a>	440
9.52.4	<a href="#">Member Data Documentation</a>	440
9.52.4.1	<a href="#">logscroll</a>	440
9.52.4.2	<a href="#">logtext</a>	440
9.52.4.3	<a href="#">sendevent</a>	441
9.53	<a href="#">Icc::EventReceived Class Reference</a>	441
9.53.1	<a href="#">Detailed Description</a>	441
9.53.2	<a href="#">Constructor &amp; Destructor Documentation</a>	441
9.53.2.1	<a href="#">EventReceived()</a>	441
9.53.3	<a href="#">Member Function Documentation</a>	442
9.53.3.1	<a href="#">_Close()</a>	442
9.53.4	<a href="#">Member Data Documentation</a>	442
9.53.4.1	<a href="#">eventid</a>	442

9.54 FCFSupport::PDFFileStructures::FontDictionary Class Reference . . . . .	443
9.54.1 Detailed Description . . . . .	444
9.54.2 Constructor & Destructor Documentation . . . . .	444
9.54.2.1 FontDictionary() . . . . .	444
9.54.2.2 ~FontDictionary() . . . . .	444
9.54.3 Member Function Documentation . . . . .	444
9.54.3.1 WriteDictionaryElements() . . . . .	445
9.54.3.2 WriteFontType() . . . . .	445
9.54.4 Member Data Documentation . . . . .	445
9.54.4.1 subType . . . . .	445
9.55 FCFSupport::PDFFileStructures::FreedObject Class Reference . . . . .	446
9.55.1 Detailed Description . . . . .	446
9.55.2 Constructor & Destructor Documentation . . . . .	446
9.55.2.1 FreedObject() . . . . .	446
9.55.2.2 ~FreedObject() . . . . .	447
9.55.3 Member Function Documentation . . . . .	447
9.55.3.1 WriteDirect() . . . . .	447
9.56 xpressnet::FunctionStatus Class Reference . . . . .	447
9.56.1 Detailed Description . . . . .	448
9.56.2 Constructor & Destructor Documentation . . . . .	449
9.56.2.1 FunctionStatus() . . . . .	449
9.56.3 Member Function Documentation . . . . .	449
9.56.3.1 Status() . . . . .	450
9.56.4 Member Data Documentation . . . . .	450
9.56.4.1 _status0 . . . . .	450
9.56.4.2 _status1 . . . . .	450
9.56.4.3 _status10 . . . . .	450
9.56.4.4 _status11 . . . . .	450



9.56.4.5	<a href="#">_status12</a>	451
9.56.4.6	<a href="#">_status2</a>	451
9.56.4.7	<a href="#">_status3</a>	451
9.56.4.8	<a href="#">_status4</a>	451
9.56.4.9	<a href="#">_status5</a>	451
9.56.4.10	<a href="#">_status6</a>	451
9.56.4.11	<a href="#">_status7</a>	452
9.56.4.12	<a href="#">_status8</a>	452
9.56.4.13	<a href="#">_status9</a>	452
9.57	<a href="#">linuxgpio::GpioInputActiveHigh Class Reference</a>	452
9.57.1	<a href="#">Detailed Description</a>	452
9.57.2	<a href="#">Constructor &amp; Destructor Documentation</a>	453
9.57.2.1	<a href="#">GpioInputActiveHigh()</a>	453
9.57.2.2	<a href="#">~GpioInputActiveHigh()</a>	454
9.57.3	<a href="#">Member Data Documentation</a>	454
9.57.3.1	<a href="#">basepin</a>	454
9.58	<a href="#">linuxgpio::GpioInputActiveLow Class Reference</a>	454
9.58.1	<a href="#">Detailed Description</a>	454
9.58.2	<a href="#">Constructor &amp; Destructor Documentation</a>	455
9.58.2.1	<a href="#">GpioInputActiveLow()</a>	455
9.58.2.2	<a href="#">~GpioInputActiveLow()</a>	456
9.58.3	<a href="#">Member Function Documentation</a>	456
9.58.3.1	<a href="#">Get()</a>	456
9.58.4	<a href="#">Member Data Documentation</a>	456
9.58.4.1	<a href="#">basepin</a>	456
9.59	<a href="#">linuxgpio::GpioOutputSafeHigh Class Reference</a>	456
9.59.1	<a href="#">Detailed Description</a>	456
9.59.2	<a href="#">Constructor &amp; Destructor Documentation</a>	457

9.59.2.1	GpioOutputSafeHigh()	457
9.59.2.2	~GpioOutputSafeHigh()	458
9.59.3	Member Data Documentation	458
9.59.3.1	basepin	458
9.60	linuxgpio::GpioOutputSafeHighInvert Class Reference	458
9.60.1	Detailed Description	458
9.60.2	Constructor & Destructor Documentation	459
9.60.2.1	GpioOutputSafeHighInvert()	459
9.60.2.2	~GpioOutputSafeHighInvert()	460
9.60.3	Member Function Documentation	460
9.60.3.1	Clr()	460
9.60.3.2	Get()	460
9.60.3.3	Set()	460
9.60.4	Member Data Documentation	460
9.60.4.1	basepin	461
9.61	linuxgpio::GpioOutputSafeLow Class Reference	461
9.61.1	Detailed Description	461
9.61.2	Constructor & Destructor Documentation	461
9.61.2.1	GpioOutputSafeLow()	462
9.61.2.2	~GpioOutputSafeLow()	462
9.61.3	Member Data Documentation	462
9.61.3.1	basepin	462
9.62	linuxgpio::GpioOutputSafeLowInverted Class Reference	463
9.62.1	Detailed Description	463
9.62.2	Constructor & Destructor Documentation	463
9.62.2.1	GpioOutputSafeLowInverted()	464
9.62.2.2	~GpioOutputSafeLowInverted()	464
9.62.3	Member Function Documentation	464

9.62.3.1	Clr()	464
9.62.3.2	Get()	465
9.62.3.3	Set()	465
9.62.4	Member Data Documentation	465
9.62.4.1	basepin	465
9.63	Icc::GridConnectMessage Class Reference	465
9.63.1	Detailed Description	466
9.63.2	Constructor & Destructor Documentation	467
9.63.2.1	GridConnectMessage()	467
9.63.3	Member Function Documentation	468
9.63.3.1	_copyCM()	468
9.63.3.2	_get_extended()	468
9.63.3.3	_get_rtr()	469
9.63.3.4	_set_extended()	469
9.63.3.5	_set_rtr()	469
9.63.3.6	setByte()	470
9.63.3.7	setHeader()	470
9.63.3.8	setHexDigit()	471
9.64	Icc::GridConnectReply Class Reference	471
9.64.1	Detailed Description	472
9.64.2	Constructor & Destructor Documentation	473
9.64.2.1	GridConnectReply()	474
9.64.3	Member Function Documentation	474
9.64.3.1	_copyGCM()	474
9.64.3.2	_get_extended()	474
9.64.3.3	_get_rtr()	475
9.64.3.4	basicFormatCheck()	475
9.64.3.5	createReply()	476

9.64.3.6	<a href="#">getByte()</a>	476
9.64.3.7	<a href="#">getHeader()</a>	476
9.64.3.8	<a href="#">getHexDigit()</a>	476
9.64.3.9	<a href="#">getNumBytes()</a>	477
9.64.3.10	<a href="#">maxSize()</a>	477
9.64.3.11	<a href="#">setData()</a>	477
9.64.3.12	<a href="#">setElement()</a>	478
9.64.4	<a href="#">Member Data Documentation</a>	478
9.64.4.1	<a href="#">_RTRoffset</a>	478
9.64.4.2	<a href="#">MAXLEN</a>	478
9.65	<a href="#">TTSupport::hash Struct Reference</a>	478
9.65.1	<a href="#">Detailed Description</a>	479
9.65.2	<a href="#">Member Function Documentation</a>	479
9.65.2.1	<a href="#">operator&gt;()</a>	479
9.66	<a href="#">CTCPanel::HiddenBlock Class Reference</a>	479
9.66.1	<a href="#">Detailed Description</a>	480
9.66.2	<a href="#">Constructor &amp; Destructor Documentation</a>	481
9.66.2.1	<a href="#">HiddenBlock()</a>	481
9.66.2.2	<a href="#">~HiddenBlock()</a>	481
9.66.3	<a href="#">Member Function Documentation</a>	481
9.66.3.1	<a href="#">_configureLabel()</a>	481
9.66.3.2	<a href="#">geti()</a>	482
9.66.3.3	<a href="#">getv()</a>	482
9.66.3.4	<a href="#">invoke()</a>	482
9.66.3.5	<a href="#">seti()</a>	482
9.66.3.6	<a href="#">setv()</a>	482
9.66.4	<a href="#">Member Data Documentation</a>	483
9.66.4.1	<a href="#">canvas</a>	483

9.66.4.2	ctcpanel	483
9.67	HTMLHelp::HTMLHelp Class Reference	483
9.67.1	Detailed Description	488
9.67.2	Constructor & Destructor Documentation	488
9.67.2.1	HTMLHelp()	488
9.67.3	Member Function Documentation	489
9.67.3.1	_Close()	489
9.67.3.2	_SBackward()	489
9.67.3.3	_SForward()	489
9.67.3.4	a()	489
9.67.3.5	back()	489
9.67.3.6	backcurrenttopic()	490
9.67.3.7	color()	490
9.67.3.8	findtopicintoc()	490
9.67.3.9	font()	490
9.67.3.10	form()	490
9.67.3.11	forward()	491
9.67.3.12	forwardcurrenttopic()	491
9.67.3.13	get_html()	491
9.67.3.14	GetInstance()	491
9.67.3.15	help()	491
9.67.3.16	helpTopic()	492
9.67.3.17	HMapend_css()	492
9.67.3.18	HMcgiDecode()	492
9.67.3.19	HMcgiMap()	492
9.67.3.20	HMcheck_tocRelative()	492
9.67.3.21	HMcurrent_tags()	493
9.67.3.22	HMdo_map()	493

9.67.3.23 HMextract_param()	493
9.67.3.24 HMgot_image()	493
9.67.3.25 HMgoto()	494
9.67.3.26 HMinit_state()	494
9.67.3.27 HMinit_win()	494
9.67.3.28 HMinput_checkbox()	494
9.67.3.29 HMinput_hidden()	494
9.67.3.30 HMinput_image()	495
9.67.3.31 HMinput_password()	495
9.67.3.32 HMinput_radio()	495
9.67.3.33 HMinput_reset()	495
9.67.3.34 HMinput_submit()	495
9.67.3.35 HMinput_text()	496
9.67.3.36 Hmlink_callback()	496
9.67.3.37 Hmlink_hit()	496
9.67.3.38 Hmlink_setup()	496
9.67.3.39 HMload_css()	497
9.67.3.40 HMmap_esc()	497
9.67.3.41 HMmap_reply()	497
9.67.3.42 HMOptimize()	497
9.67.3.43 HMparse_html()	497
9.67.3.44 HMrender()	498
9.67.3.45 HMreset_win()	498
9.67.3.46 HMset_font()	498
9.67.3.47 HMset_image()	498
9.67.3.48 HMset_indent()	499
9.67.3.49 HMset_state()	499
9.67.3.50 HMstack()	499

9.67.3.51 hmstart()	499
9.67.3.52 Hmsubmit_button()	500
9.67.3.53 Hmsubmit_form()	500
9.67.3.54 Hmsubmit_index()	500
9.67.3.55 Hmtag_a()	501
9.67.3.56 Hmtag_color()	501
9.67.3.57 Hmtag_dt()	501
9.67.3.58 Hmtag_font() [1/2]	501
9.67.3.59 Hmtag_font() [2/2]	502
9.67.3.60 Hmtag_form()	502
9.67.3.61 Hmtag_hmstart()	502
9.67.3.62 Hmtag_hr()	502
9.67.3.63 Hmtag_img()	503
9.67.3.64 Hmtag_input()	503
9.67.3.65 Hmtag_isindex()	503
9.67.3.66 Hmtag_li()	504
9.67.3.67 Hmtag_link()	504
9.67.3.68 Hmtag_menu()	504
9.67.3.69 Hmtag_ol()	504
9.67.3.70 Hmtag_option()	504
9.67.3.71 Hmtag_select()	505
9.67.3.72 Hmtag_textarea()	505
9.67.3.73 Hmtag_title()	505
9.67.3.74 Hmtag_ul()	505
9.67.3.75 Hmtest_parse()	506
9.67.3.76 Hmwent_to()	506
9.67.3.77 Hmwin_install()	506
9.67.3.78 Hmx_font()	506

9.67.3.79	HMzap_white()	507
9.67.3.80	menu()	507
9.67.3.81	nextlink()	507
9.67.3.82	prevlink()	507
9.67.3.83	pushcurrenttopic()	507
9.67.3.84	render()	508
9.67.3.85	searchbackward()	508
9.67.3.86	searchforward()	508
9.67.3.87	select()	508
9.67.3.88	setDefault()	508
9.67.4	Member Data Documentation	509
9.67.4.1	_WidgetMap	509
9.67.4.2	command	509
9.67.4.3	curtopicindex	509
9.67.4.4	defaultHelpDirectory	509
9.67.4.5	defaultHelpWindow	510
9.67.4.6	defaultTableOfContents	510
9.67.4.7	Fonts	510
9.67.4.8	helptext	510
9.67.4.9	helptext_css	510
9.67.4.10	HMalphanumeric	510
9.67.4.11	HMesc_map	511
9.67.4.12	HMevents	511
9.67.4.13	HMform_map	511
9.67.4.14	HMinert_map	511
9.67.4.15	HMlist_elements	511
9.67.4.16	HMparam_map	511
9.67.4.17	HMtag_map	511



9.67.4.18 lastsearch . . . . .	512
9.67.4.19 panes . . . . .	512
9.67.4.20 status . . . . .	512
9.67.4.21 textscroll . . . . .	512
9.67.4.22 toc . . . . .	512
9.67.4.23 toc_css . . . . .	513
9.67.4.24 tocscroll . . . . .	513
9.67.4.25 topicstack . . . . .	513
9.67.4.26 Url . . . . .	513
9.68 FCFSupport::PDFFileStructures::IndirectFloatVector Class Reference . . . . .	513
9.68.1 Detailed Description . . . . .	514
9.68.2 Constructor & Destructor Documentation . . . . .	514
9.68.2.1 IndirectFloatVector() . . . . .	514
9.68.2.2 ~IndirectFloatVector() . . . . .	514
9.68.3 Member Function Documentation . . . . .	515
9.68.3.1 WriteDirect() . . . . .	515
9.69 FCFSupport::PDFFileStructures::IndirectObject Class Reference . . . . .	515
9.69.1 Detailed Description . . . . .	516
9.69.2 Constructor & Destructor Documentation . . . . .	516
9.69.2.1 IndirectObject() . . . . .	517
9.69.2.2 ~IndirectObject() . . . . .	517
9.69.3 Member Function Documentation . . . . .	517
9.69.3.1 FileOffset() . . . . .	517
9.69.3.2 GenerationNumber() . . . . .	517
9.69.3.3 HasOffset() . . . . .	518
9.69.3.4 IncrementGenerationNumber() . . . . .	518
9.69.3.5 ObjectNumber() . . . . .	518
9.69.3.6 SetObjectNumber() . . . . .	518

9.69.3.7	WriteDirect()	519
9.69.3.8	WriteIndirectReference()	519
9.69.3.9	WriteObjectToFile()	519
9.69.4	Friends And Related Function Documentation	520
9.69.4.1	CrossReferenceTable	520
9.69.5	Member Data Documentation	520
9.69.5.1	fileOffset	520
9.69.5.2	generationNumber	520
9.69.5.3	objectNumber	520
9.69.5.4	table	521
9.70	FCFSupport::PDFFileStructures::IndirectObjectDictionary Class Reference	521
9.70.1	Detailed Description	522
9.70.2	Constructor & Destructor Documentation	522
9.70.2.1	IndirectObjectDictionary()	522
9.70.2.2	~IndirectObjectDictionary()	522
9.70.3	Member Function Documentation	522
9.70.3.1	AddIndirectObject()	522
9.70.3.2	Size()	523
9.70.3.3	WriteDictionaryElements()	523
9.70.4	Member Data Documentation	523
9.70.4.1	elements	523
9.71	FCFSupport::Industry Class Reference	524
9.71.1	Detailed Description	526
9.71.2	Constructor & Destructor Documentation	526
9.71.2.1	Industry() [1/3]	526
9.71.2.2	Industry() [2/3]	527
9.71.2.3	Industry() [3/3]	528
9.71.3	Member Function Documentation	529

9.71.3.1	<a href="#">AssignLen()</a>	529
9.71.3.2	<a href="#">CarsLen()</a>	529
9.71.3.3	<a href="#">CarsNum()</a>	529
9.71.3.4	<a href="#">DivisionControlList()</a>	529
9.71.3.5	<a href="#">EmptiesAccepted()</a>	530
9.71.3.6	<a href="#">Hazard()</a>	530
9.71.3.7	<a href="#">IncrementStatsLen()</a>	530
9.71.3.8	<a href="#">LoadsAccepted()</a>	530
9.71.3.9	<a href="#">MaxCarLen()</a>	530
9.71.3.10	<a href="#">MaxPlate()</a>	531
9.71.3.11	<a href="#">MaxWeightClass()</a>	531
9.71.3.12	<a href="#">MyMirror()</a>	531
9.71.3.13	<a href="#">MyStation()</a>	531
9.71.3.14	<a href="#">Name()</a>	531
9.71.3.15	<a href="#">NumberOfCars()</a>	532
9.71.3.16	<a href="#">operator=()</a>	532
9.71.3.17	<a href="#">Priority()</a>	532
9.71.3.18	<a href="#">Reload()</a>	532
9.71.3.19	<a href="#">StatsLen()</a>	533
9.71.3.20	<a href="#">TheCar()</a>	533
9.71.3.21	<a href="#">TrackLen()</a>	533
9.71.3.22	<a href="#">Type()</a>	533
9.71.4	<a href="#">Friends And Related Function Documentation</a>	533
9.71.4.1	<a href="#">System</a>	534
9.71.5	<a href="#">Member Data Documentation</a>	534
9.71.5.1	<a href="#">assignLen</a>	534
9.71.5.2	<a href="#">cars</a>	534
9.71.5.3	<a href="#">carsLen</a>	534

9.71.5.4 carsNum . . . . .	534
9.71.5.5 divisionControlList . . . . .	535
9.71.5.6 emptyTypes . . . . .	535
9.71.5.7 hazard . . . . .	535
9.71.5.8 loadTypes . . . . .	535
9.71.5.9 maxCarLen . . . . .	535
9.71.5.10 mirror . . . . .	536
9.71.5.11 name . . . . .	536
9.71.5.12 plate . . . . .	536
9.71.5.13 priority . . . . .	536
9.71.5.14 reload . . . . .	536
9.71.5.15 remLen . . . . .	537
9.71.5.16 station . . . . .	537
9.71.5.17 statsLen . . . . .	537
9.71.5.18 trackLen . . . . .	537
9.71.5.19 type . . . . .	537
9.71.5.20 usedLen . . . . .	538
9.71.5.21 weightclass . . . . .	538
9.72 FCFSupport::PDFFileStructures::InformationDirectory Class Reference . . . . .	538
9.72.1 Detailed Description . . . . .	539
9.72.2 Constructor & Destructor Documentation . . . . .	539
9.72.2.1 InformationDirectory() . . . . .	539
9.72.2.2 ~InformationDirectory() . . . . .	540
9.72.3 Member Function Documentation . . . . .	540
9.72.3.1 WriteDictionaryElements() . . . . .	540
9.72.4 Member Data Documentation . . . . .	540
9.72.4.1 author . . . . .	540
9.72.4.2 creator . . . . .	541

9.72.4.3	creationDate	541
9.72.4.4	keywords	541
9.72.4.5	modificationDate	541
9.72.4.6	producer	541
9.72.4.7	subject	541
9.72.4.8	title	542
9.73	Parsers::IntegerList Class Reference	542
9.73.1	Detailed Description	543
9.73.2	Constructor & Destructor Documentation	543
9.73.2.1	IntegerList()	543
9.73.3	Member Function Documentation	543
9.73.3.1	CleanUpIntegerList()	543
9.73.3.2	CopyList()	543
9.73.3.3	Element()	544
9.73.3.4	ElementP()	544
9.73.3.5	IntAppend()	544
9.73.3.6	Next() [1/2]	544
9.73.3.7	Next() [2/2]	545
9.73.4	Friends And Related Function Documentation	545
9.73.4.1	operator<<	545
9.73.4.2	TurnoutBodyElt	545
9.73.5	Member Data Documentation	545
9.73.5.1	iElt	545
9.73.5.2	next	546
9.74	CTCPanel::Lamp Class Reference	546
9.74.1	Detailed Description	546
9.74.2	Constructor & Destructor Documentation	547
9.74.2.1	Lamp()	547

9.74.2.2	<code>~Lamp()</code>	548
9.74.3	Member Function Documentation	548
9.74.3.1	<code>_configureColor()</code>	548
9.74.3.2	<code>_configureLabel()</code>	548
9.74.3.3	<code>geti()</code>	548
9.74.3.4	<code>getv()</code>	549
9.74.3.5	<code>invoke()</code>	549
9.74.3.6	<code>seti()</code>	549
9.74.3.7	<code>setv()</code>	549
9.74.4	Member Data Documentation	549
9.74.4.1	<code>canvas</code>	549
9.74.4.2	<code>ctcpanel</code>	550
9.74.4.3	<code>state</code>	550
9.75	Parsers::LayoutFile Class Reference	550
9.75.1	Detailed Description	553
9.75.2	Constructor & Destructor Documentation	553
9.75.2.1	<code>LayoutFile()</code>	553
9.75.2.2	<code>~LayoutFile()</code>	553
9.75.3	Member Function Documentation	553
9.75.3.1	<code>Angle()</code>	553
9.75.3.2	<code>CompressedEdgeCount()</code>	554
9.75.3.3	<code>CompressedEdgeLength()</code>	554
9.75.3.4	<code>CompressedEdgeNode()</code>	554
9.75.3.5	<code>CompressedGraphCircleLayout()</code>	555
9.75.3.6	<code>CompressedGraphKamadaKawaiSpring()</code>	555
9.75.3.7	<code>CompressedGraphKruskalMinimumSpanningTree()</code>	555
9.75.3.8	<code>CompressedGraphPrimMinimumSpanningTree()</code>	555
9.75.3.9	<code>CompressedNodePositionX()</code>	556

9.75.3.10 CompressedNodePositionY()	556
9.75.3.11 CompressedNodeSegments()	556
9.75.3.12 CompressGraph()	556
9.75.3.13 EdgeA()	557
9.75.3.14 EdgeIndex()	557
9.75.3.15 EdgeLength()	557
9.75.3.16 EdgeX()	557
9.75.3.17 EdgeY()	558
9.75.3.18 Emit()	558
9.75.3.19 Heads()	558
9.75.3.20 HighestNode()	558
9.75.3.21 IsCompressed()	559
9.75.3.22 IsCompressedNode()	559
9.75.3.23 IsNodeP()	559
9.75.3.24 LengthOfNode()	559
9.75.3.25 LowestNode()	559
9.75.3.26 NameOfNode()	560
9.75.3.27 NodeTurnoutGraphic()	560
9.75.3.28 NodeTurnoutRoutelist()	560
9.75.3.29 NormalActionScript()	560
9.75.3.30 NumberOfHeads()	561
9.75.3.31 NumEdges()	561
9.75.3.32 OffScript()	561
9.75.3.33 OnScript()	561
9.75.3.34 OrigX()	562
9.75.3.35 OrigY()	562
9.75.3.36 Parse()	562
9.75.3.37 ParseError()	563

9.75.3.38 ReverseActionScript()	563
9.75.3.39 Roots()	563
9.75.3.40 SenseScript()	563
9.75.3.41 SignalAspects()	564
9.75.3.42 TrackList()	564
9.75.3.43 TurnoutNumber()	564
9.75.3.44 TypeOfNode()	564
9.75.4 Member Data Documentation	564
9.75.4.1 parser	565
9.75.4.2 trackGraph	565
9.76 xpressnet::LI100Message Class Reference	565
9.76.1 Detailed Description	565
9.76.2 Constructor & Destructor Documentation	565
9.76.2.1 LI100Message()	565
9.76.3 Member Function Documentation	566
9.76.3.1 MessageType()	566
9.76.4 Member Data Documentation	566
9.76.4.1 _message_type	566
9.77 xpressnet::LI100VersionNumbers Class Reference	566
9.77.1 Detailed Description	567
9.77.2 Constructor & Destructor Documentation	567
9.77.2.1 LI100VersionNumbers()	567
9.77.3 Member Function Documentation	567
9.77.3.1 HardwareVersion()	567
9.77.3.2 SoftwareVersion()	568
9.77.4 Member Data Documentation	568
9.77.4.1 _hardware_version	568
9.77.4.2 _software_version	568



9.78	xpressnet::LI101XPressNetAddress Class Reference	568
9.78.1	Detailed Description	569
9.78.2	Constructor & Destructor Documentation	569
9.78.2.1	LI101XPressNetAddress()	569
9.78.3	Member Function Documentation	569
9.78.3.1	Address()	569
9.78.4	Member Data Documentation	569
9.78.4.1	_address	569
9.79	linuxgpio::LinuxGpio Class Reference	570
9.79.1	Detailed Description	570
9.79.2	Constructor & Destructor Documentation	571
9.79.2.1	LinuxGpio()	571
9.79.2.2	~LinuxGpio()	571
9.79.3	Member Function Documentation	571
9.79.3.1	Clr()	572
9.79.3.2	Get()	572
9.79.3.3	is_output()	572
9.79.3.4	read()	572
9.79.3.5	Set()	572
9.79.3.6	write()	572
9.79.4	Member Data Documentation	573
9.79.4.1	DIRECTIONFMT	573
9.79.4.2	EXPORT	573
9.79.4.3	UNEXPORT	573
9.79.4.4	VALUEFMT	573
9.80	xpressnet::LocomotiveAddress Class Reference	573
9.80.1	Detailed Description	574
9.80.2	Constructor & Destructor Documentation	574

9.80.2.1	LocomotiveAddress()	574
9.80.3	Member Function Documentation	575
9.80.3.1	Address()	575
9.80.3.2	AddressType()	575
9.80.4	Member Data Documentation	575
9.80.4.1	_address	575
9.80.4.2	_addressType	575
9.81	CabWidgets::LocomotiveDirection Class Reference	575
9.81.1	Detailed Description	576
9.81.2	Constructor & Destructor Documentation	577
9.81.2.1	LocomotiveDirection()	577
9.81.3	Member Function Documentation	577
9.81.3.1	_setdirection()	577
9.81.3.2	direction()	578
9.81.3.3	direction_sense()	578
9.81.3.4	invoke()	578
9.81.4	Member Data Documentation	578
9.81.4.1	_direction	578
9.81.4.2	_left	579
9.81.4.3	_right	579
9.81.4.4	currentDirection	579
9.81.4.5	forward	579
9.81.4.6	reverse	579
9.82	xpressnet::LocomotiveInformation Class Reference	579
9.82.1	Detailed Description	581
9.82.2	Constructor & Destructor Documentation	581
9.82.2.1	LocomotiveInformation()	581
9.82.3	Member Function Documentation	582

9.82.3.1	Address()	582
9.82.3.2	Address2()	583
9.82.3.3	Available()	583
9.82.3.4	Direction()	583
9.82.3.5	Function()	583
9.82.3.6	MTR()	583
9.82.3.7	Speed()	584
9.82.3.8	SpeedStepMode()	584
9.82.4	Member Data Documentation	584
9.82.4.1	_address	584
9.82.4.2	_address2	584
9.82.4.3	_available	584
9.82.4.4	_direction	585
9.82.4.5	_function0	585
9.82.4.6	_function1	585
9.82.4.7	_function10	585
9.82.4.8	_function11	585
9.82.4.9	_function12	585
9.82.4.10	_function2	586
9.82.4.11	_function3	586
9.82.4.12	_function4	586
9.82.4.13	_function5	586
9.82.4.14	_function6	586
9.82.4.15	_function7	586
9.82.4.16	_function8	587
9.82.4.17	_function9	587
9.82.4.18	_mtraddress	587
9.82.4.19	_speed	587

9.82.4.20	<code>_speedstep</code>	587
9.83	<code>CabWidgets::LocomotiveSpeed</code> Class Reference	587
9.83.1	Detailed Description	589
9.83.2	Constructor & Destructor Documentation	589
9.83.2.1	<code>LocomotiveSpeed()</code>	589
9.83.3	Member Function Documentation	590
9.83.3.1	<code>_down1()</code>	590
9.83.3.2	<code>_down10()</code>	590
9.83.3.3	<code>_setspeed()</code>	590
9.83.3.4	<code>_stop()</code>	590
9.83.3.5	<code>_up1()</code>	591
9.83.3.6	<code>_up10()</code>	591
9.83.3.7	<code>invoke()</code>	591
9.83.3.8	<code>setspeed()</code>	591
9.83.3.9	<code>speed()</code>	591
9.83.4	Member Data Documentation	591
9.83.4.1	<code>_down</code>	592
9.83.4.2	<code>_down10</code>	592
9.83.4.3	<code>_speed</code>	592
9.83.4.4	<code>_stop</code>	592
9.83.4.5	<code>_up</code>	592
9.83.4.6	<code>_up10</code>	592
9.83.4.7	<code>bar</code>	593
9.83.4.8	<code>down1</code>	593
9.83.4.9	<code>down10</code>	593
9.83.4.10	<code>leftbuttons</code>	593
9.83.4.11	<code>rightbuttons</code>	593
9.83.4.12	<code>stop</code>	593

9.83.4.13 up1 . . . . .	594
9.83.4.14 up10 . . . . .	594
9.84 FCFSupport::LogMessageCallback Class Reference . . . . .	594
9.84.1 Detailed Description . . . . .	594
9.84.2 Member Enumeration Documentation . . . . .	595
9.84.2.1 MessageType . . . . .	595
9.84.3 Constructor & Destructor Documentation . . . . .	596
9.84.3.1 LogMessageCallback() . . . . .	596
9.84.3.2 ~LogMessageCallback() . . . . .	596
9.84.4 Member Function Documentation . . . . .	596
9.84.4.1 LogMessage() . . . . .	596
9.85 FCFSupport::LQ24PrinterDevice Class Reference . . . . .	597
9.85.1 Detailed Description . . . . .	598
9.85.2 Member Enumeration Documentation . . . . .	598
9.85.2.1 ChCodes . . . . .	598
9.85.3 Constructor & Destructor Documentation . . . . .	599
9.85.3.1 LQ24PrinterDevice() . . . . .	599
9.85.3.2 ~LQ24PrinterDevice() . . . . .	599
9.85.4 Member Function Documentation . . . . .	600
9.85.4.1 ClosePrinter() . . . . .	600
9.85.4.2 NewPage() . . . . .	600
9.85.4.3 OpenPrinter() . . . . .	600
9.85.4.4 Put() . . . . .	601
9.85.4.5 PutLine() . . . . .	601
9.85.4.6 SetTypeSlant() . . . . .	602
9.85.4.7 SetTypeSpacing() . . . . .	602
9.85.4.8 SetTypeWeight() . . . . .	602
9.85.4.9 Tab() . . . . .	603

9.85.5	Member Data Documentation	603
9.85.5.1	currentColumn	603
9.85.5.2	currentColumnFraction	603
9.85.5.3	currentSlant	603
9.85.5.4	currentSpacing	604
9.85.5.5	currentWeight	604
9.85.5.6	oneColumnWidthFraction	604
9.85.5.7	printerStream	604
9.86	mainwindow Class Reference	604
9.86.1	Detailed Description	607
9.86.2	Package provided	608
9.86.3	Constructor & Destructor Documentation	608
9.86.3.1	mainwindow()	608
9.86.4	Member Function Documentation	609
9.86.4.1	buttons_add()	609
9.86.4.2	buttons_delete()	609
9.86.4.3	buttons_hide()	609
9.86.4.4	buttons_insert()	610
9.86.4.5	buttons_itemconfigure()	610
9.86.4.6	buttons_show()	610
9.86.4.7	menu_activate()	610
9.86.4.8	menu_add()	611
9.86.4.9	menu_delete()	611
9.86.4.10	menu_entrycget()	611
9.86.4.11	menu_entryconfigure()	612
9.86.4.12	menu_index()	612
9.86.4.13	menu_insert()	613
9.86.4.14	menu_invoke()	613

9.86.4.15 menu_sethelpvar()	613
9.86.4.16 menu_type()	614
9.86.4.17 setprogress()	614
9.86.4.18 setstatus()	614
9.86.4.19 showit()	614
9.86.4.20 slideout_add()	615
9.86.4.21 slideout_getframe()	615
9.86.4.22 slideout_hide()	615
9.86.4.23 slideout_issownp()	616
9.86.4.24 slideout_list()	616
9.86.4.25 slideout_reqwidth()	616
9.86.4.26 slideout_show()	616
9.86.4.27 toolbar_add()	617
9.86.4.28 toolbar_addbutton()	617
9.86.4.29 toolbar_buttoncget()	617
9.86.4.30 toolbar_buttonconfigure()	618
9.86.4.31 toolbar_hide()	618
9.86.4.32 toolbar_setbuttonstate()	618
9.86.4.33 toolbar_show()	619
9.86.5 Member Data Documentation	619
9.86.5.1 buttons	619
9.86.5.2 numtoolbars	619
9.86.5.3 panewindow	619
9.86.5.4 progress	620
9.86.5.5 scrollwindow	620
9.86.5.6 slideouts	620
9.86.5.7 status	620
9.86.5.8 toolbars	620

9.86.5.9 wipmessage . . . . .	620
9.87 azatrax::MRD Class Reference . . . . .	621
9.87.1 Detailed Description . . . . .	622
9.87.2 Member Enumeration Documentation . . . . .	623
9.87.2.1 OperatingMode_Type . . . . .	623
9.87.3 Constructor & Destructor Documentation . . . . .	623
9.87.3.1 MRD() . . . . .	623
9.87.3.2 ~MRD() . . . . .	623
9.87.4 Member Function Documentation . . . . .	624
9.87.4.1 AllowingExternalChanges() . . . . .	624
9.87.4.2 ClearExternallyChanged() . . . . .	624
9.87.4.3 DisableExternal() . . . . .	624
9.87.4.4 EnableExternal() . . . . .	625
9.87.4.5 ExternallyChanged() . . . . .	625
9.87.4.6 HasRelays() . . . . .	625
9.87.4.7 Identify_1_2() . . . . .	625
9.87.4.8 Identify_2() . . . . .	626
9.87.4.9 Latch_1() . . . . .	626
9.87.4.10 Latch_2() . . . . .	626
9.87.4.11 OperatingMode() . . . . .	626
9.87.4.12 ResetStatus() . . . . .	627
9.87.4.13 ResetStopwatch() . . . . .	627
9.87.4.14 Sense_1() . . . . .	627
9.87.4.15 Sense_2() . . . . .	627
9.87.4.16 SetChan1() . . . . .	628
9.87.4.17 SetChan2() . . . . .	628
9.87.4.18 Stopwatch() . . . . .	628
9.87.4.19 StopwatchTicking() . . . . .	629



9.87.5 Friends And Related Function Documentation . . . . .	629
9.87.5.1 Azatrax . . . . .	629
9.88 Parsers::MRRXtrkCad Class Reference . . . . .	629
9.88.1 Detailed Description . . . . .	630
9.88.2 Member Enumeration Documentation . . . . .	630
9.88.2.1 YY_MRRXtrkCad_ENUM_TOKEN . . . . .	630
9.88.3 Constructor & Destructor Documentation . . . . .	632
9.88.3.1 MRRXtrkCad() . . . . .	632
9.88.3.2 ~MRRXtrkCad() . . . . .	632
9.88.4 Member Function Documentation . . . . .	632
9.88.4.1 lookup_word() . . . . .	632
9.88.4.2 yyerror() . . . . .	632
9.88.4.3 yyerror1() . . . . .	633
9.88.4.4 yylex() . . . . .	633
9.88.4.5 yyparse() . . . . .	633
9.88.5 Member Data Documentation . . . . .	633
9.88.5.1 CurrentScale . . . . .	633
9.88.5.2 fieldflag . . . . .	633
9.88.5.3 scanEol . . . . .	634
9.88.5.4 scanToEND . . . . .	634
9.88.5.5 yychar . . . . .	634
9.88.5.6 yydebug . . . . .	634
9.88.5.7 yylloc . . . . .	634
9.88.5.8 yylval . . . . .	634
9.88.5.9 yynerrs . . . . .	634
9.89 Icc::MTIDetail Class Reference . . . . .	635
9.89.1 Detailed Description . . . . .	636
9.89.2 Constructor & Destructor Documentation . . . . .	636

9.89.2.1	MTIDetail()	636
9.89.3	Member Function Documentation	637
9.89.3.1	getHeader()	637
9.89.3.2	setHeader()	638
9.89.4	Member Data Documentation	638
9.89.4.1	ADDRESSP_MASK	638
9.89.4.2	ADDRESSP_SHIFT	638
9.89.4.3	DESTID_MASK	639
9.89.4.4	DESTID_SHIFT	639
9.89.4.5	EVENTP_MASK	639
9.89.4.6	EVENTP_SHIFT	639
9.89.4.7	MODIFIER_MASK	639
9.89.4.8	MODIFIER_SHIFT	639
9.89.4.9	mtiheader	640
9.89.4.10	PRIORITY_MASK	640
9.89.4.11	PRIORITY_SHIFT	640
9.89.4.12	SIMPLE_MASK	640
9.89.4.13	SIMPLE_SHIFT	640
9.89.4.14	SPECIAL_MASK	640
9.89.4.15	STREAMDG_MASK	641
9.89.4.16	TYPEWITHIN_MASK	641
9.89.4.17	TYPEWITHIN_SHIFT	641
9.90	lcc::MTIHeader Class Reference	641
9.90.1	Detailed Description	642
9.90.2	Constructor & Destructor Documentation	642
9.90.2.1	MTIHeader()	642
9.90.3	Member Function Documentation	643
9.90.3.1	getHeader()	643

9.90.3.2	setHeader()	643
9.90.4	Member Data Documentation	644
9.90.4.1	canheader	644
9.90.4.2	FRAMETYPE_MASK	644
9.90.4.3	FRAMETYPE_SHIFT	644
9.90.4.4	MTI_CAN_MASK	644
9.90.4.5	MTI_CAN_SHIFT	644
9.91	nce::NCE Class Reference	645
9.91.1	Detailed Description	648
9.91.2	Constructor & Destructor Documentation	648
9.91.2.1	NCE()	649
9.91.2.2	~NCE()	649
9.91.3	Member Function Documentation	649
9.91.3.1	_explodechars()	649
9.91.3.2	_readbyte()	649
9.91.3.3	_readevent()	650
9.91.3.4	_readresponse()	650
9.91.3.5	_sendMessageAndReturnResponse()	650
9.91.3.6	_timeoutevent()	650
9.91.3.7	_transmit()	651
9.91.3.8	AccessoryDecoderOperation()	651
9.91.3.9	AddLeadLocomotiveToMultiUnit()	651
9.91.3.10	AddLocomotiveToMultiUnit()	652
9.91.3.11	AddRearLocomotiveToMultiUnit()	652
9.91.3.12	AssignLoco()	653
9.91.3.13	ChangeMomentumLevel()	653
9.91.3.14	DequeuePacket()	653
9.91.3.15	DisableMain()	654

9.91.3.16 Dummy()	654
9.91.3.17 EnableMain()	654
9.91.3.18 ExecuteMacro()	654
9.91.3.19 HardReset()	655
9.91.3.20 MacroCommand()	655
9.91.3.21 NOP()	655
9.91.3.22 NormalMode()	656
9.91.3.23 OperatingModeAccessoryProgrammingByteModeWrite()	656
9.91.3.24 OperatingModeProgrammingByteModeWrite()	656
9.91.3.25 ProgramMode()	657
9.91.3.26 ReadCVInDirectMode()	657
9.91.3.27 ReadCVInPagedMode()	657
9.91.3.28 ReadFromRAM()	658
9.91.3.29 ReadOneByteFromRAM()	658
9.91.3.30 ReadRegister()	659
9.91.3.31 RemoveLocomotiveFromMultiUnit()	659
9.91.3.32 ReturnAuxiliaryInputUnit()	659
9.91.3.33 ReturnAuxiliaryInputUnitShortForm()	660
9.91.3.34 ReturnClock()	660
9.91.3.35 SetBinaryCommandEchoMode()	660
9.91.3.36 SetCabBusAddressOfUSBBoard()	661
9.91.3.37 SetClock()	661
9.91.3.38 SetClockFormat()	662
9.91.3.39 SetClockRatio()	662
9.91.3.40 SetLocomotiveFunctionsGroup1()	662
9.91.3.41 SetLocomotiveFunctionsGroup2()	663
9.91.3.42 SetLocomotiveFunctionsGroup3()	663
9.91.3.43 SetLocomotiveSpeedAndDirection()	664

9.91.3.44 SetLocoSpeedMode()	664
9.91.3.45 SetSignalAspect()	665
9.91.3.46 SoftReset()	665
9.91.3.47 SoftwareVersion()	665
9.91.3.48 StartClock()	666
9.91.3.49 StopClock()	666
9.91.3.50 Write4BytesToRAM()	666
9.91.3.51 Write8BytesToRAM()	667
9.91.3.52 WriteCVInDirectMode()	667
9.91.3.53 WriteCVInPagedMode()	667
9.91.3.54 WriteLCDLine3()	668
9.91.3.55 WriteLCDLine4()	668
9.91.3.56 WriteLCDRightLine2()	669
9.91.3.57 WriteOneByteToRAM()	669
9.91.3.58 WriteRAWPacket()	670
9.91.3.59 WriteRAWTrackPacket()	670
9.91.3.60 WriteRegister()	670
9.91.3.61 WriteToRAM()	671
9.91.3.62 WriteTwoBytesToRAM()	671
9.91.4 Member Data Documentation	672
9.91.4.1 _timeout	672
9.91.4.2 NumberOfBytesReturned	672
9.91.4.3 ttyfd	672
9.92 Icc::nid_or_null Class Reference	672
9.92.1 Detailed Description	672
9.92.2 Member Function Documentation	673
9.92.2.1 validate()	673
9.93 Parsers::TrackGraph::NodeValues Struct Reference	673

9.93.1 Detailed Description . . . . .	674
9.93.2 Constructor & Destructor Documentation . . . . .	674
9.93.2.1 NodeValues() . . . . .	675
9.93.3 Member Function Documentation . . . . .	675
9.93.3.1 Cleanup() . . . . .	675
9.93.4 Member Data Documentation . . . . .	675
9.93.4.1 angle . . . . .	675
9.93.4.2 aspectlist . . . . .	676
9.93.4.3 id . . . . .	676
9.93.4.4 length . . . . .	676
9.93.4.5 name . . . . .	676
9.93.4.6 normalactionsript . . . . .	676
9.93.4.7 numheads . . . . .	676
9.93.4.8 offscript . . . . .	677
9.93.4.9 onscript . . . . .	677
9.93.4.10 origx . . . . .	677
9.93.4.11 origy . . . . .	677
9.93.4.12 reverseactionsript . . . . .	677
9.93.4.13 sensescript . . . . .	677
9.93.4.14 tgr . . . . .	678
9.93.4.15 tpo . . . . .	678
9.93.4.16 tracklist . . . . .	678
9.93.4.17 turnoutnumber . . . . .	678
9.93.4.18 type . . . . .	678
9.94 TTSupport::Occupied Class Reference . . . . .	678
9.94.1 Detailed Description . . . . .	679
9.94.2 Constructor & Destructor Documentation . . . . .	680
9.94.2.1 Occupied() [1/2] . . . . .	680

9.94.2.2	Occupied() [2/2]	680
9.94.3	Member Function Documentation	680
9.94.3.1	From()	681
9.94.3.2	operator=()	681
9.94.3.3	Read()	681
9.94.3.4	TrainNum()	681
9.94.3.5	TrainNum2()	682
9.94.3.6	Until()	682
9.94.3.7	Write()	682
9.94.4	Friends And Related Function Documentation	682
9.94.4.1	TimeTableSystem	683
9.94.5	Member Data Documentation	683
9.94.5.1	from	683
9.94.5.2	trainnum	683
9.94.5.3	trainnum2	683
9.94.5.4	until	684
9.95	Icc::OpenLCBMessage Class Reference	684
9.95.1	Detailed Description	684
9.95.2	Constructor & Destructor Documentation	685
9.95.2.1	OpenLCBMessage()	685
9.95.3	Member Function Documentation	685
9.95.3.1	_cgetdata()	685
9.95.3.2	_configuredata()	686
9.95.3.3	toString()	686
9.95.3.4	validate()	686
9.96	Icc::OpenLCBNode Class Reference	687
9.96.1	Detailed Description	689
9.96.2	Constructor & Destructor Documentation	689

9.96.2.1	OpenLCBNode()	689
9.96.3	Member Function Documentation	690
9.96.3.1	_buildSelectTransportConstructorDialog()	690
9.96.3.2	_CancelSelectTransport()	691
9.96.3.3	_messageHandler()	691
9.96.3.4	_SelectTransport()	691
9.96.3.5	ConsumerIdentified()	691
9.96.3.6	ConsumerRangeIdentified()	692
9.96.3.7	DatagramReceivedOK()	692
9.96.3.8	DatagramRejected()	692
9.96.3.9	IdentifyConsumer()	693
9.96.3.10	IdentifyEvents()	693
9.96.3.11	IdentifyProducer()	693
9.96.3.12	LearnEvent()	694
9.96.3.13	nidlist()	694
9.96.3.14	ProduceEvent()	694
9.96.3.15	ProducerIdentified()	694
9.96.3.16	ProducerRangeIdentified()	695
9.96.3.17	ProtocolSupportRequest()	695
9.96.3.18	ReturnMySimpleNodeInfo()	695
9.96.3.19	ReturnMySupportedProtocols()	696
9.96.3.20	selectTransportConstructor()	696
9.96.3.21	SendDatagram()	696
9.96.3.22	SendInitComplete()	697
9.96.3.23	SendMyNodeVerification()	697
9.96.3.24	SendMySimpleNodeInfo()	697
9.96.3.25	SendMySupportedProtocols()	697
9.96.3.26	SendSimpleNodeInfoRequest()	698



9.96.3.27	SendSupportedProtocolsRequest()	698
9.96.3.28	SendVerifyNodeID()	698
9.96.3.29	transportConstructors()	699
9.96.4	Member Data Documentation	699
9.96.4.1	_transportConstructors	699
9.96.4.2	constructorCombo	699
9.96.4.3	protocolsupport	699
9.96.4.4	selectTransportConstructorDialog	700
9.96.4.5	simplenodeinfo	700
9.96.4.6	transport	700
9.97	Icc::OpenLCBOverTcp Class Reference	700
9.97.1	Detailed Description	701
9.97.2	Constructor & Destructor Documentation	702
9.97.2.1	OpenLCBOverTcp()	702
9.97.3	Member Function Documentation	702
9.97.3.1	_CancelOpenTransport()	702
9.97.3.2	_OpenTransport()	703
9.97.3.3	buildPortnidandhostDialog()	703
9.97.3.4	drawOptionsDialog()	703
9.97.3.5	requiredOpts()	704
9.97.3.6	sendMessage()	704
9.97.3.7	setMessageHandler()	704
9.97.4	Member Data Documentation	705
9.97.4.1	_timeout	705
9.97.4.2	datagrambuffers	705
9.97.4.3	hostLEntry	705
9.97.4.4	messagebuffers	705
9.97.4.5	messagehandler	706

9.97.4.6	mtidetail	706
9.97.4.7	nidLEntry	706
9.97.4.8	NIDPATTERN	706
9.97.4.9	portLSpin	706
9.97.4.10	portnidandhostDialog	706
9.97.4.11	sock	707
9.98	Icc::OpenLCBProtocols Class Reference	707
9.98.1	Detailed Description	707
9.98.2	Member Function Documentation	707
9.98.2.1	GetProtocolNames()	707
9.98.2.2	InsertProtocolBit()	708
9.98.2.3	ProtocolLabelString()	708
9.98.2.4	validate()	709
9.98.3	Member Data Documentation	709
9.98.3.1	bitstype	709
9.98.3.2	protocolstrings	709
9.99	OvalWidgets::OvalButton Class Reference	709
9.99.1	Detailed Description	710
9.99.2	Constructor & Destructor Documentation	710
9.99.2.1	OvalButton()	711
9.99.2.2	~OvalButton()	711
9.99.3	Member Function Documentation	711
9.99.3.1	_ConfigureText()	711
9.99.3.2	invoke()	711
9.99.4	Member Data Documentation	712
9.99.4.1	canvas	712
9.100	OvalWidgets::OvalRoundCornerRectangle Class Reference	712
9.100.1	Detailed Description	712

9.100.2 Constructor & Destructor Documentation . . . . .	713
9.100.2.1 OvalRoundCornerRectangle() . . . . .	713
9.100.2.2 ~OvalRoundCornerRectangle() . . . . .	713
9.100.3 Member Data Documentation . . . . .	713
9.100.3.1 canvas . . . . .	713
9.101 OvalWidgets::OvalScale Class Reference . . . . .	713
9.101.1 Detailed Description . . . . .	714
9.101.2 Constructor & Destructor Documentation . . . . .	715
9.101.2.1 OvalScale() . . . . .	715
9.101.2.2 ~OvalScale() . . . . .	716
9.101.3 Member Function Documentation . . . . .	716
9.101.3.1 _BaseRect() . . . . .	716
9.101.3.2 _ConfigureText() . . . . .	716
9.101.3.3 _ConfigureWL() . . . . .	717
9.101.3.4 _ConfigureXY() . . . . .	717
9.101.3.5 _MoveThumb() . . . . .	717
9.101.3.6 get() . . . . .	718
9.101.3.7 set() . . . . .	718
9.101.4 Member Data Documentation . . . . .	718
9.101.4.1 _value . . . . .	718
9.101.4.2 canvas . . . . .	718
9.102 OvalWidgets::OvalSlider Class Reference . . . . .	718
9.102.1 Detailed Description . . . . .	719
9.102.2 Constructor & Destructor Documentation . . . . .	720
9.102.2.1 OvalSlider() . . . . .	720
9.102.3 Member Function Documentation . . . . .	721
9.102.3.1 _ConfigureL() . . . . .	721
9.102.3.2 _ConfigureText() . . . . .	721

9.102.3.3 _MoveThumb()	. 721
9.102.3.4 _VerifyBitmap()	. 722
9.102.3.5 get()	. 722
9.102.3.6 set()	. 722
9.102.4 Member Data Documentation	. 723
9.102.4.1 _value	. 723
9.102.4.2 canvas	. 723
9.103OvalWidgets::OvalScrollBar Class Reference	. 723
9.103.1 Detailed Description	. 724
9.103.2 Constructor & Destructor Documentation	. 725
9.103.2.1 OvalScrollBar()	. 725
9.103.2.2 ~OvalScrollBar()	. 725
9.103.3 Member Function Documentation	. 725
9.103.3.1 _BaseRect()	. 725
9.103.3.2 _Command()	. 726
9.103.3.3 _ConfigureWL()	. 726
9.103.3.4 _ConfigureXY()	. 726
9.103.3.5 _MoveThumb()	. 727
9.103.3.6 delta()	. 727
9.103.3.7 fraction()	. 727
9.103.3.8 get()	. 728
9.103.3.9 identify()	. 728
9.103.3.10resize()	. 728
9.103.3.11set()	. 729
9.103.4 Member Data Documentation	. 729
9.103.4.1 _lastSet	. 729
9.103.4.2 canvas	. 730
9.104FCFSupport::Owner Class Reference	. 730

9.104.1 Detailed Description	731
9.104.2 Constructor & Destructor Documentation	731
9.104.2.1 Owner() [1/3]	731
9.104.2.2 Owner() [2/3]	731
9.104.2.3 Owner() [3/3]	731
9.104.2.4 ~Owner()	732
9.104.3 Member Function Documentation	732
9.104.3.1 Comment()	732
9.104.3.2 Initials()	732
9.104.3.3 Name()	733
9.104.3.4 operator=()	733
9.104.4 Member Data Documentation	733
9.104.4.1 comment	733
9.104.4.2 initials	733
9.104.4.3 name	734
9.105FCFSupport::PDFFileStructures::Page Class Reference	734
9.105.1 Detailed Description	735
9.105.2 Constructor & Destructor Documentation	735
9.105.2.1 Page()	735
9.105.2.2 ~Page()	736
9.105.3 Member Function Documentation	736
9.105.3.1 AppendStream()	736
9.105.3.2 WriteDictionaryElements()	736
9.105.4 Friends And Related Function Documentation	736
9.105.4.1 PageTree	737
9.105.5 Member Data Documentation	737
9.105.5.1 contents	737
9.105.5.2 cropBox	737

9.105.5.3 mediaBox . . . . .	737
9.105.5.4 parent . . . . .	737
9.105.5.5 resources . . . . .	737
9.106FCFSupport::PDFFileStructures::PageLabelDictionary Class Reference . . . . .	738
9.106.1 Detailed Description . . . . .	739
9.106.2 Member Enumeration Documentation . . . . .	739
9.106.2.1 NumberStyle . . . . .	739
9.106.3 Constructor & Destructor Documentation . . . . .	739
9.106.3.1 PageLabelDictionary() . . . . .	739
9.106.3.2 ~PageLabelDictionary() . . . . .	740
9.106.4 Member Function Documentation . . . . .	740
9.106.4.1 WriteDictionaryElements() . . . . .	740
9.106.5 Member Data Documentation . . . . .	740
9.106.5.1 prefix . . . . .	741
9.106.5.2 start . . . . .	741
9.106.5.3 style . . . . .	741
9.107FCFSupport::PDFFileStructures::PageLabelTree Class Reference . . . . .	741
9.107.1 Detailed Description . . . . .	742
9.107.2 Constructor & Destructor Documentation . . . . .	742
9.107.2.1 PageLabelTree() . . . . .	742
9.107.2.2 ~PageLabelTree() . . . . .	743
9.107.3 Member Function Documentation . . . . .	743
9.107.3.1 AddPageLabelDictionary() . . . . .	743
9.107.3.2 AddPageLabelTree() . . . . .	743
9.107.3.3 GetKidLimits() . . . . .	744
9.107.3.4 Size() . . . . .	744
9.107.3.5 WriteDictionaryElements() . . . . .	744
9.107.4 Member Data Documentation . . . . .	745

9.107.4.1 isRoot . . . . .	745
9.107.4.2 kids . . . . .	745
9.107.4.3 nums . . . . .	745
9.108FCFSupport::PDFFileStructures::PageTree Class Reference . . . . .	745
9.108.1 Detailed Description . . . . .	746
9.108.2 Constructor & Destructor Documentation . . . . .	746
9.108.2.1 PageTree() . . . . .	747
9.108.2.2 ~PageTree() . . . . .	747
9.108.3 Member Function Documentation . . . . .	747
9.108.3.1 AddPage() . . . . .	747
9.108.3.2 AddPageTree() . . . . .	748
9.108.3.3 WriteDictionaryElements() . . . . .	748
9.108.4 Member Data Documentation . . . . .	748
9.108.4.1 cropBox . . . . .	748
9.108.4.2 mediaBox . . . . .	749
9.108.4.3 pagenodes . . . . .	749
9.108.4.4 parent . . . . .	749
9.108.4.5 resources . . . . .	749
9.109Parsers::ParseFile Class Reference . . . . .	749
9.109.1 Detailed Description . . . . .	750
9.109.2 Constructor & Destructor Documentation . . . . .	751
9.109.2.1 ParseFile() . . . . .	751
9.109.2.2 ~ParseFile() . . . . .	751
9.109.3 Member Function Documentation . . . . .	751
9.109.3.1 Parse() . . . . .	751
9.109.3.2 ParseError() . . . . .	751
9.109.3.3 ProcessFile() . . . . .	751
9.109.3.4 SourceFile() . . . . .	752

9.109.4 Member Data Documentation . . . . .	752
9.109.4.1 buffersize . . . . .	752
9.109.4.2 errorstream . . . . .	752
9.109.4.3 fp . . . . .	752
9.109.4.4 line_buffer . . . . .	753
9.109.4.5 lp . . . . .	753
9.109.4.6 source_file . . . . .	753
9.109.4.7 source_line . . . . .	753
9.110 ParseXML Class Reference . . . . .	753
9.110.1 Detailed Description . . . . .	754
9.110.2 Constructor & Destructor Documentation . . . . .	754
9.110.2.1 ParseXML() . . . . .	754
9.110.3 Member Function Documentation . . . . .	756
9.110.3.1 _characterdata() . . . . .	756
9.110.3.2 _elementend() . . . . .	756
9.110.3.3 _elementstart() . . . . .	756
9.110.3.4 displayTree() . . . . .	757
9.110.3.5 validate() . . . . .	757
9.110.4 Member Data Documentation . . . . .	757
9.110.4.1 nodeStack . . . . .	758
9.110.4.2 rootnode . . . . .	758
9.111 FCFSupport::PathName Class Reference . . . . .	758
9.111.1 Detailed Description . . . . .	759
9.111.2 Constructor & Destructor Documentation . . . . .	759
9.111.2.1 PathName() [1/4] . . . . .	760
9.111.2.2 PathName() [2/4] . . . . .	760
9.111.2.3 PathName() [3/4] . . . . .	760
9.111.2.4 PathName() [4/4] . . . . .	760



9.111.2.5 ~PathName()	. 761
9.111.3 Member Function Documentation	. 761
9.111.3.1 Dirname()	. 761
9.111.3.2 Extension()	. 761
9.111.3.3 FullPath()	. 761
9.111.3.4 operator+() [1/2]	. 761
9.111.3.5 operator+() [2/2]	. 762
9.111.3.6 operator+=() [1/2]	. 762
9.111.3.7 operator+=() [2/2]	. 762
9.111.3.8 operator<()	. 763
9.111.3.9 operator<=()	. 763
9.111.3.10operator=() [1/2]	. 763
9.111.3.11operator=() [2/2]	. 764
9.111.3.12operator==()	. 764
9.111.3.13operator>()	. 764
9.111.3.14operator>=()	. 765
9.111.3.15PathSeparator()	. 765
9.111.3.16SameDirectory()	. 765
9.111.3.17Split()	. 766
9.111.3.18Tail()	. 766
9.111.4 Member Data Documentation	. 766
9.111.4.1 pathname	. 766
9.112TTSupport::PathName Class Reference	. 766
9.112.1 Detailed Description	. 768
9.112.2 Constructor & Destructor Documentation	. 768
9.112.2.1 PathName() [1/4]	. 768
9.112.2.2 PathName() [2/4]	. 768
9.112.2.3 PathName() [3/4]	. 769

9.112.2.4 PathName() [ 4 / 4 ] . . . . .	769
9.112.2.5 ~PathName() . . . . .	769
9.112.3 Member Function Documentation . . . . .	769
9.112.3.1 Dirname() . . . . .	769
9.112.3.2 Extension() . . . . .	770
9.112.3.3 FullPath() . . . . .	770
9.112.3.4 operator+() [ 1 / 2 ] . . . . .	770
9.112.3.5 operator+() [ 2 / 2 ] . . . . .	770
9.112.3.6 operator+=() [ 1 / 2 ] . . . . .	771
9.112.3.7 operator+=() [ 2 / 2 ] . . . . .	771
9.112.3.8 operator<() . . . . .	771
9.112.3.9 operator<=() . . . . .	772
9.112.3.10 operator=() [ 1 / 2 ] . . . . .	772
9.112.3.11 operator=() [ 2 / 2 ] . . . . .	772
9.112.3.12 operator==( ) . . . . .	773
9.112.3.13 operator>() . . . . .	773
9.112.3.14 operator>=() . . . . .	773
9.112.3.15 PathSeparator() . . . . .	774
9.112.3.16 SameDirectory() . . . . .	774
9.112.3.17 Split() . . . . .	774
9.112.3.18 Tail() . . . . .	774
9.112.4 Member Data Documentation . . . . .	774
9.112.4.1 pathname . . . . .	775
9.113 FCFSupport::PauseCallback Class Reference . . . . .	775
9.113.1 Detailed Description . . . . .	775
9.113.2 Constructor & Destructor Documentation . . . . .	775
9.113.2.1 PauseCallback() . . . . .	775
9.113.2.2 ~PauseCallback() . . . . .	776

9.113.3 Member Function Documentation . . . . .	776
9.113.3.1 Pause() . . . . .	776
9.114FCFSupport::PDFFileStructures::PDFNameArray Class Reference . . . . .	776
9.114.1 Detailed Description . . . . .	777
9.114.2 Constructor & Destructor Documentation . . . . .	777
9.114.2.1 PDFNameArray() . . . . .	777
9.114.2.2 ~PDFNameArray() . . . . .	777
9.115FCFSupport::PDFPrinterDevice Class Reference . . . . .	777
9.115.1 Detailed Description . . . . .	779
9.115.2 Constructor & Destructor Documentation . . . . .	779
9.115.2.1 PDFPrinterDevice() . . . . .	779
9.115.2.2 ~PDFPrinterDevice() . . . . .	780
9.115.3 Member Function Documentation . . . . .	780
9.115.3.1 ClosePrinter() . . . . .	780
9.115.3.2 CreateNewPage() . . . . .	780
9.115.3.3 CreateNewStream() . . . . .	781
9.115.3.4 NewPage() . . . . .	781
9.115.3.5 OpenPrinter() . . . . .	781
9.115.3.6 Put() . . . . .	781
9.115.3.7 PutLine() . . . . .	782
9.115.3.8 SetTypeSlant() . . . . .	782
9.115.3.9 SetTypeSpacing() . . . . .	782
9.115.3.10SetTypeWeight() . . . . .	783
9.115.3.11Tab() . . . . .	783
9.115.4 Member Data Documentation . . . . .	783
9.115.4.1 crossReferenceTable . . . . .	783
9.115.4.2 currentColumn . . . . .	784
9.115.4.3 currentColumnFraction . . . . .	784

9.115.4.4 currentFontName . . . . .	784
9.115.4.5 currentPage . . . . .	784
9.115.4.6 currentStream . . . . .	784
9.115.4.7 horizontalScaling . . . . .	784
9.115.4.8 info . . . . .	785
9.115.4.9 lines . . . . .	785
9.115.4.10 maxLines . . . . .	785
9.115.4.11 needPage . . . . .	785
9.115.4.12 pageTreeRoot . . . . .	785
9.115.4.13 partline . . . . .	785
9.115.4.14 printerStream . . . . .	786
9.115.4.15 rootDictionary . . . . .	786
9.115.4.16 title . . . . .	786
9.116FCFSupport::PDFFileStructures::PDFStream Class Reference . . . . .	786
9.116.1 Detailed Description . . . . .	787
9.116.2 Constructor & Destructor Documentation . . . . .	787
9.116.2.1 PDFStream() . . . . .	787
9.116.2.2 ~PDFStream() . . . . .	787
9.116.3 Member Function Documentation . . . . .	787
9.116.3.1 WriteDirect() . . . . .	788
9.117Parsers::TrackGraph::Point Struct Reference . . . . .	788
9.117.1 Detailed Description . . . . .	788
9.117.2 Member Data Documentation . . . . .	788
9.117.2.1 x . . . . .	788
9.117.2.2 y . . . . .	789
9.118Parsers::TurnoutBodyElt::Pos Struct Reference . . . . .	789
9.118.1 Detailed Description . . . . .	789
9.118.2 Member Data Documentation . . . . .	789

9.118.2.1 x	789
9.118.2.2 y	790
9.119Parsers::BezierBodyElt::Pos Struct Reference	790
9.119.1 Detailed Description	790
9.119.2 Member Data Documentation	790
9.119.2.1 x	790
9.119.2.2 y	791
9.120Parsers::CornuBodyElt::Pos Struct Reference	791
9.120.1 Detailed Description	791
9.120.2 Member Data Documentation	791
9.120.2.1 x	791
9.120.2.2 y	792
9.121FCFSupport::PostScriptPrinterDevice Class Reference	792
9.121.1 Detailed Description	793
9.121.2 Constructor & Destructor Documentation	793
9.121.2.1 PostScriptPrinterDevice()	793
9.121.2.2 ~PostScriptPrinterDevice()	794
9.121.3 Member Function Documentation	794
9.121.3.1 ClosePrinter()	794
9.121.3.2 NewPage()	794
9.121.3.3 OpenPrinter()	795
9.121.3.4 PSQuote()	795
9.121.3.5 Put()	795
9.121.3.6 PutLine()	796
9.121.3.7 PutPageHeader()	796
9.121.3.8 SetTypeSlant()	796
9.121.3.9 SetTypeSpacing()	797
9.121.3.10SetTypeWeight()	797

9.121.3.11Tab()	797
9.121.4 Member Data Documentation	798
9.121.4.1 lines	798
9.121.4.2 maxLines	798
9.121.4.3 needPageHeader	798
9.121.4.4 pages	798
9.121.4.5 partline	798
9.121.4.6 printerStream	799
9.121.4.7 title	799
9.122FCFSupport::PDFFileStructures::PostScriptStandardType1FontDictionary Class Reference	799
9.122.1 Detailed Description	800
9.122.2 Constructor & Destructor Documentation	800
9.122.2.1 PostScriptStandardType1FontDictionary()	800
9.122.2.2 ~PostScriptStandardType1FontDictionary()	800
9.123FCFSupport::PrinterDevice Class Reference	801
9.123.1 Detailed Description	802
9.123.2 Member Enumeration Documentation	802
9.123.2.1 PageSize	802
9.123.2.2 TypeSlant	803
9.123.2.3 TypeSpacing	803
9.123.2.4 TypeWeight	803
9.123.3 Constructor & Destructor Documentation	804
9.123.3.1 PrinterDevice()	804
9.123.3.2 ~PrinterDevice()	804
9.123.4 Member Function Documentation	804
9.123.4.1 ClosePrinter()	804
9.123.4.2 IsOpenP()	805
9.123.4.3 NewPage()	805

9.123.4.4 OpenPrinter()	806
9.123.4.5 PrinterPageSize()	806
9.123.4.6 Put() [1/3]	806
9.123.4.7 Put() [2/3]	807
9.123.4.8 Put() [3/3]	807
9.123.4.9 PutLine()	807
9.123.4.10 SetTypeSlant()	808
9.123.4.11 SetTypeSpacing()	808
9.123.4.12 SetTypeWeight()	809
9.123.4.13 Tab()	809
9.123.5 Member Data Documentation	809
9.123.5.1 isOpenP	809
9.123.5.2 pageSize	810
9.124CTCPanel::PushButton Class Reference	810
9.124.1 Detailed Description	810
9.124.2 Constructor & Destructor Documentation	811
9.124.2.1 PushButton()	811
9.124.2.2 ~PushButton()	812
9.124.3 Member Function Documentation	812
9.124.3.1 _configureLabel()	812
9.124.3.2 geti()	812
9.124.3.3 getv()	812
9.124.3.4 invoke()	812
9.124.3.5 seti()	813
9.124.3.6 setv()	813
9.124.4 Member Data Documentation	813
9.124.4.1 canvas	813
9.124.4.2 ctcpnl	813

9.125raildriver::RaildriverClient Class Reference . . . . .	813
9.125.1 Detailed Description . . . . .	814
9.125.2 Constructor & Destructor Documentation . . . . .	815
9.125.2.1 RaildriverClient() . . . . .	815
9.125.2.2 ~RaildriverClient() . . . . .	815
9.125.3 Member Function Documentation . . . . .	815
9.125.3.1 _poller() . . . . .	815
9.125.3.2 _readevent() . . . . .	815
9.125.3.3 clear() . . . . .	815
9.125.3.4 leds() . . . . .	815
9.125.3.5 mask() . . . . .	816
9.125.3.6 speaker() . . . . .	816
9.125.4 Member Data Documentation . . . . .	816
9.125.4.1 pollid . . . . .	816
9.125.4.2 socket . . . . .	817
9.126RaildriverIO Class Reference . . . . .	817
9.126.1 Detailed Description . . . . .	821
9.126.2 Tcl Package Provided . . . . .	821
9.126.3 Library Provided . . . . .	821
9.126.4 Member Enumeration Documentation . . . . .	821
9.126.4.1 Eventcodes . . . . .	821
9.126.4.2 Eventmask_bits . . . . .	822
9.126.5 Constructor & Destructor Documentation . . . . .	823
9.126.5.1 RaildriverIO() [1/2] . . . . .	823
9.126.5.2 ~RaildriverIO() . . . . .	823
9.126.5.3 RaildriverIO() [2/2] . . . . .	824
9.126.6 Member Function Documentation . . . . .	824
9.126.6.1 GetAlert() . . . . .	824



9.126.6.2 GetAutoBrake()	824
9.126.6.3 GetBailOff()	824
9.126.6.4 GetBell()	825
9.126.6.5 GetBlueButton1()	825
9.126.6.6 GetBlueButton10()	825
9.126.6.7 GetBlueButton11()	825
9.126.6.8 GetBlueButton12()	825
9.126.6.9 GetBlueButton13()	826
9.126.6.10GetBlueButton14()	826
9.126.6.11GetBlueButton15()	826
9.126.6.12GetBlueButton16()	826
9.126.6.13GetBlueButton17()	826
9.126.6.14GetBlueButton18()	827
9.126.6.15GetBlueButton19()	827
9.126.6.16GetBlueButton2()	827
9.126.6.17GetBlueButton20()	827
9.126.6.18GetBlueButton21()	827
9.126.6.19GetBlueButton22()	828
9.126.6.20GetBlueButton23()	828
9.126.6.21GetBlueButton24()	828
9.126.6.22GetBlueButton25()	828
9.126.6.23GetBlueButton26()	828
9.126.6.24GetBlueButton27()	829
9.126.6.25GetBlueButton28()	829
9.126.6.26GetBlueButton3()	829
9.126.6.27GetBlueButton4()	829
9.126.6.28GetBlueButton5()	829
9.126.6.29GetBlueButton6()	830

9.126.6.30GetBlueButton7()	830
9.126.6.31GetBlueButton8()	830
9.126.6.32GetBlueButton9()	830
9.126.6.33GetEBrakeDown()	830
9.126.6.34GetEBrakeUp()	831
9.126.6.35GetHeadlight()	831
9.126.6.36GetIndependBrake()	831
9.126.6.37GetPanDown()	831
9.126.6.38GetPanLeft()	831
9.126.6.39GetPanRight()	832
9.126.6.40GetPantograph()	832
9.126.6.41GetPanUp()	832
9.126.6.42GetProductCodeId()	832
9.126.6.43GetRangeDown()	832
9.126.6.44GetRangeUp()	833
9.126.6.45GetReverser()	833
9.126.6.46GetSand()	833
9.126.6.47GetThrottle()	833
9.126.6.48GetWhistleDown()	833
9.126.6.49GetWhistleUp()	834
9.126.6.50GetWiper()	834
9.126.6.51GetZoomUp()	834
9.126.6.52GetZoopDown()	834
9.126.6.53ReadInputs()	834
9.126.6.54SetLEDS()	835
9.126.6.55SpeakerOff()	835
9.126.6.56SpeakerOn()	835
9.126.7 Member Data Documentation	835

9.126.7.1 AutoBrake . . . . .	835
9.126.7.2 BailOff . . . . .	836
9.126.7.3 Digital1 . . . . .	836
9.126.7.4 Digital2 . . . . .	836
9.126.7.5 Digital3 . . . . .	836
9.126.7.6 Digital4 . . . . .	836
9.126.7.7 Digital5 . . . . .	836
9.126.7.8 Digital6 . . . . .	837
9.126.7.9 Headlight . . . . .	837
9.126.7.10IndependBrake . . . . .	837
9.126.7.11LEDCommand . . . . .	837
9.126.7.12PIEngineering . . . . .	837
9.126.7.13ProductCodeId . . . . .	838
9.126.7.14RailDriverModernDesktop . . . . .	838
9.126.7.15RDInput . . . . .	838
9.126.7.16driverdev . . . . .	838
9.126.7.17ReportBuffer . . . . .	839
9.126.7.18Reverser . . . . .	839
9.126.7.19SpeakerCommand . . . . .	839
9.126.7.20theBytes . . . . .	839
9.126.7.21Throttle . . . . .	839
9.126.7.22Wiper . . . . .	839
9.127FCFSupport::PDFFileStructures::Rectangle Class Reference . . . . .	840
9.127.1 Detailed Description . . . . .	841
9.127.2 Constructor & Destructor Documentation . . . . .	841
9.127.2.1 Rectangle() . . . . .	841
9.127.2.2 ~Rectangle() . . . . .	841
9.127.3 Member Function Documentation . . . . .	842

9.127.3.1 WriteDirect()	842
9.127.3.2 X1()	842
9.127.3.3 X2()	842
9.127.3.4 Y1()	842
9.127.3.5 Y2()	843
9.127.4 Member Data Documentation	843
9.127.4.1 x1	843
9.127.4.2 x2	843
9.127.4.3 y1	843
9.127.4.4 y2	843
9.128FCFSupport::PDFFileStructures::ResourceDictionary Class Reference	844
9.128.1 Detailed Description	845
9.128.2 Constructor & Destructor Documentation	845
9.128.2.1 ResourceDictionary()	845
9.128.2.2 ~ResourceDictionary()	846
9.128.3 Member Function Documentation	846
9.128.3.1 AddColorSpace()	846
9.128.3.2 AddExternalGraphicsState()	846
9.128.3.3 AddFont()	847
9.128.3.4 AddPattern()	847
9.128.3.5 AddProcSet()	847
9.128.3.6 AddProperties()	848
9.128.3.7 AddShading()	848
9.128.3.8 AddXObject()	848
9.128.3.9 WriteDictionaryElements()	849
9.128.4 Member Data Documentation	849
9.128.4.1 colorSpace	849
9.128.4.2 extGState	849

9.128.4.3 font . . . . .	850
9.128.4.4 pattern . . . . .	850
9.128.4.5 procSets . . . . .	850
9.128.4.6 properties . . . . .	850
9.128.4.7 shading . . . . .	850
9.128.4.8 xObject . . . . .	850
9.129Parsers::RouteVec Struct Reference . . . . .	851
9.129.1 Detailed Description . . . . .	851
9.129.2 Member Data Documentation . . . . .	851
9.129.2.1 positionName . . . . .	851
9.129.2.2 posList . . . . .	851
9.129.2.3 routeLength . . . . .	852
9.130Satellite Class Reference . . . . .	852
9.130.1 Detailed Description . . . . .	852
9.130.2 Constructor & Destructor Documentation . . . . .	853
9.130.2.1 Satellite() . . . . .	853
9.130.2.2 ~Satellite() . . . . .	853
9.130.3 Member Function Documentation . . . . .	853
9.130.3.1 remoteeval() . . . . .	853
9.130.4 Member Data Documentation . . . . .	854
9.130.4.1 socket . . . . .	854
9.131CTCPanel::SchLabel Class Reference . . . . .	854
9.131.1 Detailed Description . . . . .	855
9.131.2 Constructor & Destructor Documentation . . . . .	856
9.131.2.1 SchLabel() . . . . .	856
9.131.2.2 ~SchLabel() . . . . .	856
9.131.3 Member Function Documentation . . . . .	856
9.131.3.1 _configureColor() . . . . .	856

9.131.3.2 _configureLabel()	857
9.131.3.3 geti()	857
9.131.3.4 getv()	857
9.131.3.5 invoke()	857
9.131.3.6 seti()	857
9.131.3.7 setv()	858
9.131.4 Member Data Documentation	858
9.131.4.1 canvas	858
9.131.4.2 ctcpnl	858
9.132CTCPanel::ScissorCrossover Class Reference	858
9.132.1 Detailed Description	859
9.132.2 Constructor & Destructor Documentation	860
9.132.2.1 ScissorCrossover()	860
9.132.2.2 ~ScissorCrossover()	860
9.132.3 Member Function Documentation	860
9.132.3.1 _configureLabel()	861
9.132.3.2 geti()	861
9.132.3.3 getv()	861
9.132.3.4 invoke()	861
9.132.3.5 seti()	861
9.132.3.6 setv()	861
9.132.4 Member Data Documentation	862
9.132.4.1 canvas	862
9.132.4.2 ctcpnl	862
9.132.4.3 state	862
9.133ScrollTabNotebook Class Reference	862
9.133.1 Detailed Description	865
9.133.2 Constructor & Destructor Documentation	865

9.133.2.1 ScrollTabNotebook()	865
9.133.3 Member Function Documentation	866
9.133.3.1 _compute_height()	866
9.133.3.2 _compute_width()	866
9.133.3.3 _Configure()	866
9.133.3.4 _draw_arrows()	867
9.133.3.5 _draw_page()	867
9.133.3.6 _get_x_page()	867
9.133.3.7 _highlight()	868
9.133.3.8 _redraw()	868
9.133.3.9 _resize()	868
9.133.3.10 _select()	868
9.133.3.11 _test_page()	869
9.133.3.12 _themeChanged()	869
9.133.3.13 _themeChanged_()	869
9.133.3.14 _xview()	870
9.133.3.15 add()	870
9.133.3.16 compute_size()	871
9.133.3.17 forget()	871
9.133.3.18 get3dcolor()	871
9.133.3.19 index()	871
9.133.3.20 insert()	872
9.133.3.21 isee()	872
9.133.3.22 select()	873
9.133.3.23 tab()	873
9.133.3.24 tabs()	874
9.133.4 Member Data Documentation	874
9.133.4.1 _clientRow	874

9.133.4.2 _hpage . . . . .	875
9.133.4.3 _left . . . . .	875
9.133.4.4 _paddingtype . . . . .	875
9.133.4.5 _radiustype . . . . .	875
9.133.4.6 _right . . . . .	875
9.133.4.7 _tabrow . . . . .	875
9.133.4.8 _tabsides . . . . .	876
9.133.4.9 _textid . . . . .	876
9.133.4.10 _warrow . . . . .	876
9.133.4.11 _wpage . . . . .	876
9.133.4.12 base . . . . .	876
9.133.4.13 bg . . . . .	876
9.133.4.14 bg . . . . .	877
9.133.4.15 left . . . . .	877
9.133.4.16 pages . . . . .	877
9.133.4.17 pages_opts . . . . .	877
9.133.4.18 realized . . . . .	877
9.133.4.19 right . . . . .	877
9.133.4.20 select . . . . .	878
9.133.4.21 tabrow . . . . .	878
9.133.4.22 tabs . . . . .	878
9.134Parsers::SegPos Struct Reference . . . . .	878
9.134.1 Detailed Description . . . . .	878
9.134.2 Member Data Documentation . . . . .	879
9.134.2.1 x . . . . .	879
9.134.2.2 y . . . . .	879
9.135Parsers::SegVector Struct Reference . . . . .	879
9.135.1 Detailed Description . . . . .	880



9.135.2 Member Enumeration Documentation . . . . .	880
9.135.2.1 GrType . . . . .	880
9.135.3 Member Data Documentation . . . . .	881
9.135.3.1 ang0 . . . . .	881
9.135.3.2 ang1 . . . . .	881
9.135.3.3 angle . . . . .	881
9.135.3.4 ePos1 . . . . .	881
9.135.3.5 ePos2 . . . . .	881
9.135.3.6 gPos1 . . . . .	882
9.135.3.7 gPos2 . . . . .	882
9.135.3.8 L . . . . .	882
9.135.3.9 len0 . . . . .	882
9.135.3.10 len1 . . . . .	882
9.135.3.11 length . . . . .	882
9.135.3.12 R . . . . .	883
9.135.3.13 radius . . . . .	883
9.135.3.14 gType . . . . .	883
9.136 CabWidgets::SelectLocomotive Class Reference . . . . .	883
9.136.1 Detailed Description . . . . .	884
9.136.2 Constructor & Destructor Documentation . . . . .	884
9.136.2.1 SelectLocomotive() . . . . .	884
9.136.3 Member Function Documentation . . . . .	885
9.136.3.1 _addnewloco() . . . . .	885
9.136.3.2 _trimList() . . . . .	885
9.136.3.3 currentLocomotive() . . . . .	885
9.136.3.4 invoke() . . . . .	886
9.136.4 Member Data Documentation . . . . .	886
9.136.4.1 lf . . . . .	886

9.136.4.2 locoList . . . . .	886
9.137Icc::SendEvent Class Reference . . . . .	886
9.137.1 Detailed Description . . . . .	887
9.137.2 Constructor & Destructor Documentation . . . . .	887
9.137.2.1 SendEvent() . . . . .	887
9.137.3 Member Function Documentation . . . . .	887
9.137.3.1 _Close() . . . . .	887
9.137.3.2 _Send() . . . . .	888
9.137.4 Member Data Documentation . . . . .	888
9.137.4.1 eventid . . . . .	888
9.138xpressnet::ServiceModeResponse Class Reference . . . . .	888
9.138.1 Detailed Description . . . . .	889
9.138.2 Constructor & Destructor Documentation . . . . .	889
9.138.2.1 ServiceModeResponse() . . . . .	889
9.138.3 Member Function Documentation . . . . .	889
9.138.3.1 CV() . . . . .	889
9.138.3.2 Data() . . . . .	889
9.138.3.3 ServiceMode() . . . . .	890
9.138.4 Member Data Documentation . . . . .	890
9.138.4.1 _cv . . . . .	890
9.138.4.2 _data . . . . .	890
9.138.4.3 _service_mode . . . . .	890
9.139FCFSupport::ShowBannerCallback Class Reference . . . . .	890
9.139.1 Detailed Description . . . . .	891
9.139.2 Constructor & Destructor Documentation . . . . .	891
9.139.2.1 ShowBannerCallback() . . . . .	891
9.139.2.2 ~ShowBannerCallback() . . . . .	891
9.139.3 Member Function Documentation . . . . .	891

9.139.3.1 ShowBanner()	892
9.140CTCPanel::Signal Class Reference	892
9.140.1 Detailed Description	893
9.140.2 Constructor & Destructor Documentation	893
9.140.2.1 Signal()	894
9.140.2.2 ~Signal()	894
9.140.3 Member Function Documentation	894
9.140.3.1 _configureLabel()	894
9.140.3.2 _SchematicDrawOval()	894
9.140.3.3 _SchematicDrawThinLine()	895
9.140.3.4 _VerifyHeads()	895
9.140.3.5 geti()	896
9.140.3.6 getv()	896
9.140.3.7 invoke()	896
9.140.3.8 seti()	896
9.140.3.9 setv()	896
9.140.4 Member Data Documentation	897
9.140.4.1 aspect	897
9.140.4.2 canvas	897
9.140.4.3 ctcpnl	897
9.141CTCPanel::SIGPlate Class Reference	897
9.141.1 Detailed Description	898
9.141.2 Constructor & Destructor Documentation	899
9.141.2.1 SIGPlate()	899
9.141.2.2 ~SIGPlate()	899
9.141.3 Member Function Documentation	900
9.141.3.1 _configureLabel()	900
9.141.3.2 geti()	900

9.141.3.3	getv()	900
9.141.3.4	invoke()	900
9.141.3.5	seti()	900
9.141.3.6	setv()	901
9.141.4	Member Data Documentation	901
9.141.4.1	_PlatePolygon	901
9.141.4.2	canvas	901
9.141.4.3	ctcpanel	901
9.142	SimpleDOMEElement Class Reference	902
9.142.1	Detailed Description	903
9.142.2	Constructor & Destructor Documentation	903
9.142.2.1	SimpleDOMEElement()	903
9.142.3	Member Function Documentation	903
9.142.3.1	_formattrlist()	904
9.142.3.2	_quoteXML()	904
9.142.3.3	addchild()	904
9.142.3.4	attribute()	905
9.142.3.5	children()	905
9.142.3.6	data()	905
9.142.3.7	display()	905
9.142.3.8	getElementsById()	906
9.142.3.9	getElementsByTagName()	906
9.142.3.10	getParent()	907
9.142.3.11	isChild()	907
9.142.3.12	length()	907
9.142.3.13	removeChild()	908
9.142.3.14	setAttribute()	908
9.142.3.15	setdata()	908

9.142.3.10 validate()	908
9.142.4 Member Data Documentation	909
9.142.4.1 _children	909
9.142.4.2 _data	909
9.143CTCPanel::SingleSlip Class Reference	909
9.143.1 Detailed Description	910
9.143.2 Constructor & Destructor Documentation	911
9.143.2.1 SingleSlip()	911
9.143.2.2 ~SingleSlip()	911
9.143.3 Member Function Documentation	911
9.143.3.1 _configureLabel()	912
9.143.3.2 geti()	912
9.143.3.3 getv()	912
9.143.3.4 invoke()	912
9.143.3.5 seti()	912
9.143.3.6 setv()	912
9.143.4 Member Data Documentation	913
9.143.4.1 canvas	913
9.143.4.2 ctcpnl	913
9.143.4.3 state	913
9.144azatrax::SL2 Class Reference	913
9.144.1 Detailed Description	915
9.144.2 Constructor & Destructor Documentation	915
9.144.2.1 SL2()	915
9.144.2.2 ~SL2()	916
9.144.3 Member Function Documentation	916
9.144.3.1 Input_1_Enabled()	916
9.144.3.2 Input_2_Enabled()	916

9.144.3.3 Input_3_Enabled()	916
9.144.3.4 Input_4_Enabled()	917
9.144.3.5 Motor_1_Direction()	917
9.144.3.6 Motor_1_State()	917
9.144.3.7 Motor_2_Direction()	917
9.144.3.8 Motor_2_State()	918
9.144.3.9 OutputRelayInputControl()	918
9.144.3.10Sense_1()	918
9.144.3.11Sense_2()	919
9.144.3.12Sense_3()	919
9.144.3.13Sense_4()	919
9.144.3.14SetQ1negQ2pos()	919
9.144.3.15SetQ1posQ2neg()	920
9.144.3.16SetQ1Q2open()	920
9.144.3.17SetQ3negQ4pos()	920
9.144.3.18SetQ3posQ4neg()	920
9.144.3.19SetQ3Q4open()	920
9.144.4 Friends And Related Function Documentation	921
9.144.4.1 Azatrax	921
9.145pressnet::SoftwareVersion Class Reference	921
9.145.1 Detailed Description	921
9.145.2 Constructor & Destructor Documentation	922
9.145.2.1 SoftwareVersion()	922
9.145.3 Member Function Documentation	922
9.145.3.1 CommandStationTypeCode()	922
9.145.3.2 Major()	922
9.145.3.3 Minor()	922
9.145.4 Member Data Documentation	923

9.145.4.1 <code>_command_station_type</code> . . . . .	923
9.145.4.2 <code>_major</code> . . . . .	923
9.145.4.3 <code>_minor</code> . . . . .	923
9.146 <code>splash</code> Class Reference . . . . .	923
9.146.1 Detailed Description . . . . .	924
9.146.2 Package provided . . . . .	925
9.146.3 Constructor & Destructor Documentation . . . . .	925
9.146.3.1 <code>splash()</code> . . . . .	925
9.146.4 Member Function Documentation . . . . .	925
9.146.4.1 <code>CheckColor()</code> . . . . .	925
9.146.4.2 <code>CheckImage()</code> . . . . .	926
9.146.4.3 <code>enableClickDestroy()</code> . . . . .	926
9.146.4.4 <code>hide()</code> . . . . .	926
9.146.4.5 <code>show()</code> . . . . .	926
9.146.4.6 <code>update()</code> . . . . .	926
9.146.5 Member Data Documentation . . . . .	927
9.146.5.1 <code>currentProgress</code> . . . . .	927
9.146.5.2 <code>header</code> . . . . .	927
9.146.5.3 <code>icon</code> . . . . .	927
9.146.5.4 <code>image</code> . . . . .	927
9.146.5.5 <code>progressBar</code> . . . . .	928
9.146.5.6 <code>status</code> . . . . .	928
9.146.5.7 <code>title</code> . . . . .	928
9.147 <code>azatrax::SR4</code> Class Reference . . . . .	928
9.147.1 Detailed Description . . . . .	930
9.147.2 Constructor & Destructor Documentation . . . . .	930
9.147.2.1 <code>SR4()</code> . . . . .	930
9.147.2.2 <code>~SR4()</code> . . . . .	931

9.147.3 Member Function Documentation . . . . .	931
9.147.3.1 BlinkRelays() . . . . .	931
9.147.3.2 Input_1_Enabled() . . . . .	931
9.147.3.3 Input_2_Enabled() . . . . .	932
9.147.3.4 Input_3_Enabled() . . . . .	932
9.147.3.5 Input_4_Enabled() . . . . .	932
9.147.3.6 OutputRelayInputControl() . . . . .	932
9.147.3.7 PulseRelays() . . . . .	933
9.147.3.8 Q1_State() . . . . .	933
9.147.3.9 Q2_State() . . . . .	934
9.147.3.10Q3_State() . . . . .	934
9.147.3.11Q4_State() . . . . .	934
9.147.3.12RelaysOff() . . . . .	934
9.147.3.13RelaysOn() . . . . .	935
9.147.3.14Sense_1_Latch() . . . . .	935
9.147.3.15Sense_1_Live() . . . . .	935
9.147.3.16Sense_2_Latch() . . . . .	936
9.147.3.17Sense_2_Live() . . . . .	936
9.147.3.18Sense_3_Latch() . . . . .	936
9.147.3.19Sense_3_Live() . . . . .	936
9.147.3.20Sense_4_Latch() . . . . .	937
9.147.3.21Sense_4_Live() . . . . .	937
9.147.4 Friends And Related Function Documentation . . . . .	937
9.147.4.1 Azatrax . . . . .	937
9.148azatrax::Azatrax::StateDataPacket Struct Reference . . . . .	937
9.148.1 Detailed Description . . . . .	938
9.148.2 Member Data Documentation . . . . .	938
9.148.2.1 commandEcho . . . . .	938



9.148.2.2 endOfData . . . . .	939
9.148.2.3 operatingMode . . . . .	939
9.148.2.4 packetCount . . . . .	939
9.148.2.5 reserved . . . . .	939
9.148.2.6 status1 . . . . .	939
9.148.2.7 status2 . . . . .	940
9.148.2.8 status3 . . . . .	940
9.148.2.9 status4 . . . . .	940
9.148.2.10stopwatchHours . . . . .	940
9.148.2.11stopwatchMinutes . . . . .	941
9.149FCFSupport::Station Class Reference . . . . .	941
9.149.1 Detailed Description . . . . .	942
9.149.2 Constructor & Destructor Documentation . . . . .	942
9.149.2.1 Station() [1/3] . . . . .	942
9.149.2.2 Station() [2/3] . . . . .	942
9.149.2.3 Station() [3/3] . . . . .	943
9.149.2.4 ~Station() . . . . .	943
9.149.3 Member Function Documentation . . . . .	943
9.149.3.1 AppendIndustry() . . . . .	943
9.149.3.2 Comment() . . . . .	944
9.149.3.3 MyDivision() . . . . .	944
9.149.3.4 Name() . . . . .	944
9.149.3.5 NumberOfIndustries() . . . . .	944
9.149.3.6 operator=() . . . . .	944
9.149.3.7 TheIndustry() . . . . .	945
9.149.4 Friends And Related Function Documentation . . . . .	945
9.149.4.1 System . . . . .	945
9.149.5 Member Data Documentation . . . . .	945

9.149.5.1 comment . . . . .	945
9.149.5.2 division . . . . .	946
9.149.5.3 industries . . . . .	946
9.149.5.4 name . . . . .	946
9.150TTSupport::Station Class Reference . . . . .	946
9.150.1 Detailed Description . . . . .	948
9.150.2 Constructor & Destructor Documentation . . . . .	948
9.150.2.1 Station() [1/2] . . . . .	948
9.150.2.2 Station() [2/2] . . . . .	948
9.150.2.3 ~Station() . . . . .	949
9.150.3 Member Function Documentation . . . . .	949
9.150.3.1 AddStorageTrack() . . . . .	949
9.150.3.2 DuplicateStationIndex() . . . . .	949
9.150.3.3 FindStorageTrack() . . . . .	949
9.150.3.4 FindTrackTrainsStoredOn() . . . . .	950
9.150.3.5 FirstStorageTrack() . . . . .	950
9.150.3.6 LastStorageTrack() . . . . .	950
9.150.3.7 Name() . . . . .	950
9.150.3.8 NumberOfStorageTracks() . . . . .	951
9.150.3.9 operator=() . . . . .	951
9.150.3.10Read() . . . . .	951
9.150.3.11SetDuplicateStationIndex() . . . . .	951
9.150.3.12SMile() . . . . .	952
9.150.3.13Write() . . . . .	952
9.150.4 Member Data Documentation . . . . .	952
9.150.4.1 duplicateStationIndex . . . . .	952
9.150.4.2 name . . . . .	952
9.150.4.3 smile . . . . .	953

9.150.4.4 storageTracks . . . . .	953
9.151FCFSupport::SwitchListElement::StationOrIndustry Union Reference . . . . .	953
9.151.1 Detailed Description . . . . .	953
9.151.2 Member Data Documentation . . . . .	953
9.151.2.1 industry . . . . .	954
9.151.2.2 station . . . . .	954
9.152FCFSupport::Train::StationOrIndustry Union Reference . . . . .	954
9.152.1 Detailed Description . . . . .	954
9.152.2 Member Data Documentation . . . . .	954
9.152.2.1 industry . . . . .	955
9.152.2.2 station . . . . .	955
9.153TTSupport::StationTimes Class Reference . . . . .	955
9.153.1 Detailed Description . . . . .	956
9.153.2 Constructor & Destructor Documentation . . . . .	956
9.153.2.1 StationTimes() [1/2] . . . . .	956
9.153.2.2 StationTimes() [2/2] . . . . .	956
9.153.3 Member Function Documentation . . . . .	957
9.153.3.1 Arrival() . . . . .	957
9.153.3.2 Departure() . . . . .	957
9.153.3.3 Flag() . . . . .	957
9.153.3.4 operator=() . . . . .	957
9.153.4 Member Data Documentation . . . . .	958
9.153.4.1 arrival . . . . .	958
9.153.4.2 departure . . . . .	958
9.153.4.3 flag . . . . .	958
9.154azatrax::SR4::status1_union Union Reference . . . . .	958
9.154.1 Detailed Description . . . . .	959
9.154.2 Member Data Documentation . . . . .	959

9.154.2.1 Q1_state . . . . .	959
9.154.2.2 Q2_state . . . . .	959
9.154.2.3 Q3_state . . . . .	960
9.154.2.4 Q4_state . . . . .	960
9.154.2.5 reserved . . . . .	960
9.154.2.6 theBits . . . . .	960
9.154.2.7 theByte . . . . .	960
9.155azatrax::SL2::status1_union Union Reference . . . . .	961
9.155.1 Detailed Description . . . . .	961
9.155.2 Member Data Documentation . . . . .	961
9.155.2.1 motor_1_direction . . . . .	961
9.155.2.2 motor_1_state . . . . .	962
9.155.2.3 motor_2_direction . . . . .	962
9.155.2.4 motor_2_state . . . . .	962
9.155.2.5 reserved . . . . .	962
9.155.2.6 theBits . . . . .	962
9.155.2.7 theByte . . . . .	963
9.156azatrax::MRD::status1_union Union Reference . . . . .	963
9.156.1 Detailed Description . . . . .	963
9.156.2 Member Data Documentation . . . . .	963
9.156.2.1 latch_1 . . . . .	964
9.156.2.2 latch_2 . . . . .	964
9.156.2.3 modtype . . . . .	964
9.156.2.4 reserved . . . . .	964
9.156.2.5 sense_1 . . . . .	964
9.156.2.6 sense_2 . . . . .	965
9.156.2.7 theBits . . . . .	965
9.156.2.8 theByte . . . . .	965

9.157azatrax::MRD::status2_union Union Reference . . . . .	965
9.157.1 Detailed Description . . . . .	966
9.157.2 Member Data Documentation . . . . .	966
9.157.2.1 allowExternalChanges . . . . .	966
9.157.2.2 externallyChanged . . . . .	966
9.157.2.3 reserved . . . . .	966
9.157.2.4 resetStatus . . . . .	966
9.157.2.5 stopwatchTicking . . . . .	967
9.157.2.6 theBits . . . . .	967
9.157.2.7 theByte . . . . .	967
9.158azatrax::SR4::status2_union Union Reference . . . . .	967
9.158.1 Detailed Description . . . . .	968
9.158.2 Member Data Documentation . . . . .	968
9.158.2.1 reserved . . . . .	968
9.158.2.2 sense_1 . . . . .	968
9.158.2.3 sense_2 . . . . .	968
9.158.2.4 sense_3 . . . . .	968
9.158.2.5 sense_4 . . . . .	969
9.158.2.6 theBits . . . . .	969
9.158.2.7 theByte . . . . .	969
9.159azatrax::SL2::status2_union Union Reference . . . . .	969
9.159.1 Detailed Description . . . . .	970
9.159.2 Member Data Documentation . . . . .	970
9.159.2.1 reserved . . . . .	970
9.159.2.2 sense_1 . . . . .	970
9.159.2.3 sense_2 . . . . .	970
9.159.2.4 sense_3 . . . . .	970
9.159.2.5 sense_4 . . . . .	971

9.159.2.6 theBits . . . . .	971
9.159.2.7 theByte . . . . .	971
9.160azatrax::SR4::status3_union Union Reference . . . . .	971
9.160.1 Detailed Description . . . . .	972
9.160.2 Member Data Documentation . . . . .	972
9.160.2.1 input_1_enabled . . . . .	972
9.160.2.2 input_2_enabled . . . . .	972
9.160.2.3 input_3_enabled . . . . .	972
9.160.2.4 input_4_enabled . . . . .	972
9.160.2.5 reserved . . . . .	973
9.160.2.6 theBits . . . . .	973
9.160.2.7 theByte . . . . .	973
9.161azatrax::SL2::status3_union Union Reference . . . . .	973
9.161.1 Detailed Description . . . . .	974
9.161.2 Member Data Documentation . . . . .	974
9.161.2.1 input_1_enabled . . . . .	974
9.161.2.2 input_2_enabled . . . . .	974
9.161.2.3 input_3_enabled . . . . .	974
9.161.2.4 input_4_enabled . . . . .	974
9.161.2.5 reserved . . . . .	975
9.161.2.6 theBits . . . . .	975
9.161.2.7 theByte . . . . .	975
9.162TTSupport::Stop Class Reference . . . . .	975
9.162.1 Detailed Description . . . . .	977
9.162.2 Member Enumeration Documentation . . . . .	977
9.162.2.1 FlagType . . . . .	977
9.162.3 Constructor & Destructor Documentation . . . . .	978
9.162.3.1 Stop() [1/2] . . . . .	978

9.162.3.2 Stop() [2/2]	978
9.162.3.3 ~Stop()	978
9.162.4 Member Function Documentation	979
9.162.4.1 AddNote()	979
9.162.4.2 Departure()	979
9.162.4.3 Flag()	979
9.162.4.4 Layover()	980
9.162.4.5 Note()	980
9.162.4.6 NumberOfNotes()	980
9.162.4.7 operator=()	980
9.162.4.8 Read()	981
9.162.4.9 RemoveNote()	981
9.162.4.10SetCab()	981
9.162.4.11SetLayover()	982
9.162.4.12SetStorageTrackName()	982
9.162.4.13StationIndex()	982
9.162.4.14StorageTrackName()	983
9.162.4.15TheCab()	983
9.162.4.16Write()	983
9.162.5 Member Data Documentation	983
9.162.5.1 cab	983
9.162.5.2 flag	984
9.162.5.3 layover	984
9.162.5.4 notes	984
9.162.5.5 stationindex	984
9.162.5.6 storageTrackName	984
9.163TTSupport::StorageTrack Class Reference	985
9.163.1 Detailed Description	986

9.163.2 Constructor & Destructor Documentation . . . . .	986
9.163.2.1 StorageTrack() [1/2] . . . . .	986
9.163.2.2 ~StorageTrack() . . . . .	986
9.163.2.3 StorageTrack() [2/2] . . . . .	987
9.163.3 Member Function Documentation . . . . .	987
9.163.3.1 FindOccupied() . . . . .	987
9.163.3.2 FirstOccupied() . . . . .	987
9.163.3.3 IncludesTime() . . . . .	987
9.163.3.4 LastOccupied() . . . . .	988
9.163.3.5 Name() . . . . .	988
9.163.3.6 operator=() . . . . .	988
9.163.3.7 Read() . . . . .	988
9.163.3.8 RemovedStoredTrain() . . . . .	989
9.163.3.9 SetName() . . . . .	989
9.163.3.10StoreTrain() . . . . .	989
9.163.3.11UpdateStoredTrain() . . . . .	990
9.163.3.12UpdateStoredTrain2() . . . . .	990
9.163.3.13UpdateStoredTrainArrival() . . . . .	991
9.163.3.14UpdateStoredTrainDeparture() . . . . .	991
9.163.3.15UsedTimeRange() . . . . .	991
9.163.3.16Write() . . . . .	992
9.163.4 Member Data Documentation . . . . .	992
9.163.4.1 name . . . . .	992
9.163.4.2 occupations . . . . .	992
9.164CTCPanel::StraightBlock Class Reference . . . . .	992
9.164.1 Detailed Description . . . . .	993
9.164.2 Constructor & Destructor Documentation . . . . .	994
9.164.2.1 StraightBlock() . . . . .	994



9.164.2.2 ~StraightBlock()	995
9.164.3 Member Function Documentation	995
9.164.3.1 _configureLabel()	995
9.164.3.2 geti()	995
9.164.3.3 getv()	995
9.164.3.4 invoke()	996
9.164.3.5 seti()	996
9.164.3.6 setv()	996
9.164.4 Member Data Documentation	996
9.164.4.1 canvas	996
9.164.4.2 ctcpnl	996
9.165CTCPnl::StubYard Class Reference	997
9.165.1 Detailed Description	997
9.165.2 Constructor & Destructor Documentation	998
9.165.2.1 StubYard()	998
9.165.2.2 ~StubYard()	999
9.165.3 Member Function Documentation	999
9.165.3.1 _configureLabel()	999
9.165.3.2 geti()	999
9.165.3.3 getv()	999
9.165.3.4 invoke()	1000
9.165.3.5 seti()	1000
9.165.3.6 setv()	1000
9.165.4 Member Data Documentation	1000
9.165.4.1 _StubYard_Poly	1000
9.165.4.2 canvas	1001
9.165.4.3 ctcpnl	1001
9.166CTCPnl::Switch Class Reference	1001

9.166.1 Detailed Description . . . . .	1002
9.166.2 Constructor & Destructor Documentation . . . . .	1003
9.166.2.1 Switch() . . . . .	1003
9.166.2.2 ~Switch() . . . . .	1003
9.166.3 Member Function Documentation . . . . .	1003
9.166.3.1 _configureLabel() . . . . .	1003
9.166.3.2 geti() . . . . .	1004
9.166.3.3 getv() . . . . .	1004
9.166.3.4 invoke() . . . . .	1004
9.166.3.5 seti() . . . . .	1004
9.166.3.6 setv() . . . . .	1004
9.166.4 Member Data Documentation . . . . .	1005
9.166.4.1 canvas . . . . .	1005
9.166.4.2 ctcpnl . . . . .	1005
9.166.4.3 state . . . . .	1005
9.167FCFSupport::SwitchList Class Reference . . . . .	1005
9.167.1 Detailed Description . . . . .	1007
9.167.2 Constructor & Destructor Documentation . . . . .	1007
9.167.2.1 SwitchList() . . . . .	1007
9.167.2.2 ~SwitchList() . . . . .	1007
9.167.3 Member Function Documentation . . . . .	1007
9.167.3.1 AddSwitchListElement() [1/2] . . . . .	1007
9.167.3.2 AddSwitchListElement() [2/2] . . . . .	1008
9.167.3.3 DiscardSwitchList() . . . . .	1008
9.167.3.4 LimitCars() . . . . .	1008
9.167.3.5 NextSwitchListForCarAndIndustry() . . . . .	1009
9.167.3.6 operator[]() [1/2] . . . . .	1009
9.167.3.7 operator[]() [2/2] . . . . .	1009

9.167.3.8 PickCarEq()	1009
9.167.3.9 PickIndex()	1010
9.167.3.10 PickLocationEq()	1010
9.167.3.11 PickTrainEq()	1010
9.167.3.12 ResetLastIndex()	1011
9.167.3.13 ResetSwitchList()	1011
9.167.4 Friends And Related Function Documentation	1011
9.167.4.1 operator<<	1011
9.167.5 Member Data Documentation	1011
9.167.5.1 lastIndex	1011
9.167.5.2 limitCars	1012
9.167.5.3 pickIndex	1012
9.167.5.4 theList	1012
9.168 FCFSupport::SwitchListElement Class Reference	1012
9.168.1 Detailed Description	1014
9.168.2 Constructor & Destructor Documentation	1014
9.168.2.1 SwitchListElement() [1/4]	1014
9.168.2.2 SwitchListElement() [2/4]	1014
9.168.2.3 SwitchListElement() [3/4]	1014
9.168.2.4 SwitchListElement() [4/4]	1015
9.168.3 Member Function Documentation	1015
9.168.3.1 DropStopEQ()	1016
9.168.3.2 DropStopIndustry()	1016
9.168.3.3 DropStopStation()	1016
9.168.3.4 LastTrain()	1016
9.168.3.5 operator=()	1016
9.168.3.6 PickCar()	1017
9.168.3.7 PickLocation()	1017

9.168.3.8 PickTrain()	1017
9.168.4 Friends And Related Function Documentation	1017
9.168.4.1 System	1017
9.168.5 Member Data Documentation	1018
9.168.5.1 dropStop	1018
9.168.5.2 lastTrain	1018
9.168.5.3 pickCar	1018
9.168.5.4 pickLoc	1018
9.168.5.5 pickTrain	1019
9.169CTCPanel::SWPlate Class Reference	1019
9.169.1 Detailed Description	1020
9.169.2 Constructor & Destructor Documentation	1021
9.169.2.1 SWPlate()	1021
9.169.2.2 ~SWPlate()	1021
9.169.3 Member Function Documentation	1021
9.169.3.1 _configureLabel()	1021
9.169.3.2 geti()	1021
9.169.3.3 getv()	1022
9.169.3.4 invoke()	1022
9.169.3.5 seti()	1022
9.169.3.6 setv()	1022
9.169.4 Member Data Documentation	1023
9.169.4.1 _PlatePolygon	1023
9.169.4.2 canvas	1023
9.169.4.3 ctcpnl	1023
9.170FCFSupport::System Class Reference	1023
9.170.1 Detailed Description	1035
9.170.2 Member Enumeration Documentation	1036

9.170.2.1 CarLocationType . . . . .	1036
9.170.2.2 CarTypeReport . . . . .	1036
9.170.3 Constructor & Destructor Documentation . . . . .	1036
9.170.3.1 System() [1/2] . . . . .	1036
9.170.3.2 System() [2/2] . . . . .	1037
9.170.3.3 ~System() . . . . .	1037
9.170.4 Member Function Documentation . . . . .	1037
9.170.4.1 AddCar() . . . . .	1037
9.170.4.2 AddOwner() . . . . .	1038
9.170.4.3 CarAssignment() . . . . .	1038
9.170.4.4 CarMovements() . . . . .	1039
9.170.4.5 CarsAtDest() . . . . .	1039
9.170.4.6 CarsAtDest_CarsInTransit() . . . . .	1039
9.170.4.7 CarsAtWorkBench() . . . . .	1039
9.170.4.8 CarsFile() . . . . .	1039
9.170.4.9 CarsInTransit() . . . . .	1040
9.170.4.10CarsMoved() . . . . .	1040
9.170.4.11CarsMovedMore() . . . . .	1040
9.170.4.12CarsMovedOnce() . . . . .	1040
9.170.4.13CarsMovedThree() . . . . .	1040
9.170.4.14CarsMovedTwice() . . . . .	1041
9.170.4.15CarsNotMoved() . . . . .	1041
9.170.4.16CarTypesFile() . . . . .	1041
9.170.4.17CarTypesOrder() . . . . .	1041
9.170.4.18CarTypesOrderIndex() . . . . .	1042
9.170.4.19DeleteAllExistingCars() . . . . .	1042
9.170.4.20FindCarInCarVector() . . . . .	1042
9.170.4.21FindDivisionByIndex() . . . . .	1042

9.170.4.22FindDivisionBySymbol()	1043
9.170.4.23FindDivisionIndex()	1043
9.170.4.24FindIndustry()	1043
9.170.4.25FindIndustryByIndex()	1044
9.170.4.26FindIndustryByName()	1044
9.170.4.27FindIndustryIndex()	1044
9.170.4.28FindStationByName()	1045
9.170.4.29FindStationIndex()	1045
9.170.4.30FindTrainByIndex()	1045
9.170.4.31FindTrainByName()	1046
9.170.4.32FirstCarType()	1046
9.170.4.33FirstDivision()	1046
9.170.4.34FirstIndustry()	1047
9.170.4.35FirstOwner()	1047
9.170.4.36FirstStation()	1047
9.170.4.37FirstTrain()	1047
9.170.4.38FixedRouteMirrorCheck()	1047
9.170.4.39FormatDutyTime()	1048
9.170.4.40GetCarStatus()	1048
9.170.4.41GetIndustryCarCounts()	1048
9.170.4.42GlobStringMatch()	1049
9.170.4.43GlobStringMatchHelper()	1049
9.170.4.44IndRipTrack()	1049
9.170.4.45IndRipTrackConst()	1050
9.170.4.46IndScrapYard()	1050
9.170.4.47IndustriesFile()	1050
9.170.4.48IndustryIndex()	1050
9.170.4.49IndustryTakesCar()	1051

9.170.4.50InternalRunOneTrain()	1051
9.170.4.51LastCarType()	1051
9.170.4.52LastDivision()	1052
9.170.4.53LastIndustry()	1052
9.170.4.54LastOwner()	1052
9.170.4.55LastStation()	1052
9.170.4.56LastTrain()	1052
9.170.4.57LoadCarFile()	1052
9.170.4.58LoadStatsFile()	1053
9.170.4.59LogCarPickup()	1053
9.170.4.60NextShift()	1053
9.170.4.61NumberOfCars()	1054
9.170.4.62NumberOfDivisions()	1054
9.170.4.63NumberOfIndustries()	1054
9.170.4.64NumberOfStations()	1054
9.170.4.65NumberOfTrains()	1055
9.170.4.66OrdersFile()	1055
9.170.4.67OtherCarOkForTrain()	1055
9.170.4.68OwnersFile()	1055
9.170.4.69PrintAllCarTypes()	1056
9.170.4.70PrintAllLists()	1056
9.170.4.71PrintAlpha()	1056
9.170.4.72PrintAnalysisHeader()	1056
9.170.4.73PrintAtwice()	1057
9.170.4.74PrintCarHeading()	1057
9.170.4.75PrintCarTypesHeader()	1057
9.170.4.76PrintCarTypesSummaryHeader()	1058
9.170.4.77PrintDashedLine()	1058

9.170.4.78PrintDispatch()	1058
9.170.4.79PrintDispatcher()	1058
9.170.4.80Printem()	1059
9.170.4.81PrintFormFeed()	1059
9.170.4.82PrintIndustryHeader()	1059
9.170.4.83PrintList()	1060
9.170.4.84PrintLocCommon()	1060
9.170.4.85PrintLocOneIndustry()	1060
9.170.4.86PrintLtwice()	1060
9.170.4.87PrintOneAnalysis()	1061
9.170.4.88PrintOneCarInfo()	1061
9.170.4.89PrintOneCarLocation()	1061
9.170.4.90PrintOneCarType()	1062
9.170.4.91PrintOneIndustry()	1062
9.170.4.92PrintSystemBanner()	1063
9.170.4.93PrintTrainLoc()	1063
9.170.4.94PrintTrainOrderHeader()	1064
9.170.4.95PrintTrainOrders()	1064
9.170.4.96PrintYards()	1064
9.170.4.97RanAllTrains()	1065
9.170.4.98Random()	1065
9.170.4.99Randomize()	1065
9.170.4.100ReadCarTypes()	1065
9.170.4.101ReadDivisions()	1066
9.170.4.102ReadGroupLimit()	1066
9.170.4.103ReadIndustries()	1066
9.170.4.104ReadOwners()	1067
9.170.4.105ReadStations()	1067



9.170.4.106	ReadTrainOrders()	1067
9.170.4.107	ReadTrains()	1068
9.170.4.108	ReLoadCarFile()	1068
9.170.4.109	ReportAnalysis()	1068
9.170.4.110	ReportCarLocations()	1069
9.170.4.111	ReportCarOwners()	1069
9.170.4.112	ReportCars()	1070
9.170.4.113	ReportCarsNotMoved()	1070
9.170.4.114	ReportCarTypes()	1070
9.170.4.115	ReportIndustries()	1071
9.170.4.116	ReportLocAll()	1071
9.170.4.117	ReportLocDivision()	1072
9.170.4.118	ReportLocIndustry()	1072
9.170.4.119	ReportLocStation()	1073
9.170.4.120	ReportTrains()	1073
9.170.4.121	ResetIndustryStats()	1073
9.170.4.122	RestartLoop()	1074
9.170.4.123	RunAllTrains()	1074
9.170.4.124	RunBoxMoves()	1074
9.170.4.125	RunOneLocal()	1075
9.170.4.126	RunOneManifest()	1075
9.170.4.127	RunOnePassenger()	1076
9.170.4.128	RunOneTrain()	1076
9.170.4.129	SaveCars()	1077
9.170.4.130	SearchForCarIndexesByNumber()	1077
9.170.4.131	SearchForIndustryPattern()	1078
9.170.4.132	SearchForTrainPattern()	1078
9.170.4.133	SessionNumber()	1078

9.170.4.133	SetPrintAlpha()	1079
9.170.4.134	SetPrintAtwice()	1080
9.170.4.135	SetPrintDispatch()	1080
9.170.4.136	SetPrintem()	1080
9.170.4.137	SetPrintList()	1081
9.170.4.138	SetPrintLtwice()	1081
9.170.4.140	SetPrintYards()	1081
9.170.4.141	ShiftNumber()	1082
9.170.4.142	ShowCarMovements()	1082
9.170.4.143	ShowCarsInDivision()	1082
9.170.4.144	ShowCarsNotMoved()	1083
9.170.4.145	ShowTrainCars()	1083
9.170.4.146	ShowTrainTotals()	1084
9.170.4.147	ShowUnassignedCars()	1084
9.170.4.148	SkipCommentsGets()	1084
9.170.4.149	Split()	1085
9.170.4.150	StatsFile()	1085
9.170.4.151	StatsPeriod()	1085
9.170.4.152	StringToInt()	1086
9.170.4.153	StringToIntRange()	1086
9.170.4.154	SystemFile()	1086
9.170.4.155	SystemName()	1087
9.170.4.156	TheCar()	1087
9.170.4.157	TheCarGroup()	1087
9.170.4.158	TheCarType()	1088
9.170.4.159	TheDivision()	1088
9.170.4.160	TheIndustry()	1088
9.170.4.161	TheOwner()	1089

9.170.4.162	TheStation()	1089
9.170.4.163	Today()	1089
9.170.4.164	TotalCars()	1090
9.170.4.165	TotalShifts()	1090
9.170.4.166	TrainByIndex()	1090
9.170.4.167	TrainByName()	1090
9.170.4.168	TrainCarPickupCheck()	1091
9.170.4.169	TrainDropAllCars()	1091
9.170.4.170	TrainDropOneCar()	1092
9.170.4.171	TrainIndex()	1092
9.170.4.172	TrainLocalDrops()	1093
9.170.4.173	TrainLocalOriginate()	1093
9.170.4.174	TrainLocalPickups()	1094
9.170.4.175	TrainManifestDrops()	1094
9.170.4.176	TrainManifestPickups()	1095
9.170.4.177	TrainPickupOneCar()	1095
9.170.4.178	TrainPrintConsistSummary()	1096
9.170.4.179	TrainPrintFinalSummary()	1096
9.170.4.180	TrainPrintTown()	1097
9.170.4.181	TrainsFile()	1097
9.170.4.182	Trim()	1097
9.170.4.183	UpperCase()	1098
9.170.4.184	WriteOneCarToDisk()	1098
9.170.5	Member Data Documentation	1098
9.170.5.1	carDest	1098
9.170.5.2	carGroups	1099
9.170.5.3	carMovements	1099
9.170.5.4	cars	1099

9.170.5.5 carsAtDest . . . . .	1099
9.170.5.6 carsAtDest_carsInTransit . . . . .	1099
9.170.5.7 carsAtWorkBench . . . . .	1100
9.170.5.8 carsFile . . . . .	1100
9.170.5.9 carsInTransit . . . . .	1100
9.170.5.10 carsMoved . . . . .	1100
9.170.5.11 carsMovedMore . . . . .	1100
9.170.5.12 carsMovedOnce . . . . .	1101
9.170.5.13 carsMovedThree . . . . .	1101
9.170.5.14 carsMovedTwice . . . . .	1101
9.170.5.15 carsNotMoved . . . . .	1101
9.170.5.16 carTypes . . . . .	1101
9.170.5.17 carTypesFile . . . . .	1102
9.170.5.18 carTypesOrder . . . . .	1102
9.170.5.19 curDiv . . . . .	1102
9.170.5.20 deliver . . . . .	1102
9.170.5.21 divisions . . . . .	1102
9.170.5.22 ndScrapYard . . . . .	1103
9.170.5.23 industries . . . . .	1103
9.170.5.24 industriesFile . . . . .	1103
9.170.5.25 messageBuffer . . . . .	1103
9.170.5.26 numberCars . . . . .	1103
9.170.5.27 ordersFile . . . . .	1104
9.170.5.28 originYard . . . . .	1104
9.170.5.29 owners . . . . .	1104
9.170.5.30 ownersFile . . . . .	1104
9.170.5.31 printAlpha . . . . .	1104
9.170.5.32 printAtwice . . . . .	1105

9.170.5.33	printDispatch	1105
9.170.5.34	printem	1105
9.170.5.35	printList	1105
9.170.5.36	printTwice	1105
9.170.5.37	printYards	1106
9.170.5.38	ranAllTrains	1106
9.170.5.39	sessionNumber	1106
9.170.5.40	shiftNumber	1106
9.170.5.41	stations	1106
9.170.5.42	statsFile	1107
9.170.5.43	statsPeriod	1107
9.170.5.44	switchList	1107
9.170.5.45	systemFile	1107
9.170.5.46	systemName	1107
9.170.5.47	totalLoads	1108
9.170.5.48	totalPickups	1108
9.170.5.49	totalRevenueTons	1108
9.170.5.50	totalShifts	1108
9.170.5.51	totalTons	1108
9.170.5.52	trainEmpties	1109
9.170.5.53	trainIndex	1109
9.170.5.54	trainLastLocation	1109
9.170.5.55	trainLength	1109
9.170.5.56	trainLoads	1109
9.170.5.57	trainLongest	1110
9.170.5.58	trainPrintOK	1110
9.170.5.59	trains	1110
9.170.5.60	trainsFile	1110

9.170.5.61trainTons . . . . .	1110
9.170.5.62wayFreight . . . . .	1111
9.170.5.63whitespace . . . . .	1111
9.171FCFSupport::TextPrinterDevice Class Reference . . . . .	1111
9.171.1 Detailed Description . . . . .	1112
9.171.2 Constructor & Destructor Documentation . . . . .	1112
9.171.2.1 TextPrinterDevice() . . . . .	1112
9.171.2.2 ~TextPrinterDevice() . . . . .	1113
9.171.3 Member Function Documentation . . . . .	1113
9.171.3.1 ClosePrinter() . . . . .	1113
9.171.3.2 NewPage() . . . . .	1113
9.171.3.3 OpenPrinter() . . . . .	1114
9.171.3.4 Put() . . . . .	1114
9.171.3.5 PutLine() . . . . .	1114
9.171.3.6 Tab() . . . . .	1115
9.171.4 Member Data Documentation . . . . .	1115
9.171.4.1 currentColumn . . . . .	1115
9.171.4.2 printerStream . . . . .	1115
9.172CTCPanel::ThreeWaySW Class Reference . . . . .	1115
9.172.1 Detailed Description . . . . .	1116
9.172.2 Constructor & Destructor Documentation . . . . .	1117
9.172.2.1 ThreeWaySW() . . . . .	1117
9.172.2.2 ~ThreeWaySW() . . . . .	1118
9.172.3 Member Function Documentation . . . . .	1118
9.172.3.1 _configureLabel() . . . . .	1118
9.172.3.2 geti() . . . . .	1118
9.172.3.3 getv() . . . . .	1118
9.172.3.4 invoke() . . . . .	1119

9.172.3.5 seti()	1119
9.172.3.6 setv()	1119
9.172.4 Member Data Documentation	1119
9.172.4.1 canvas	1119
9.172.4.2 ctcpnl	1120
9.172.4.3 state	1120
9.173CTCPanel::ThroughYard Class Reference	1120
9.173.1 Detailed Description	1121
9.173.2 Constructor & Destructor Documentation	1121
9.173.2.1 ThroughYard()	1122
9.173.2.2 ~ThroughYard()	1122
9.173.3 Member Function Documentation	1122
9.173.3.1 _configureLabel()	1122
9.173.3.2 geti()	1122
9.173.3.3 invoke()	1123
9.173.3.4 seti()	1123
9.173.3.5 setv()	1123
9.173.4 Member Data Documentation	1123
9.173.4.1 _ThroughYard_Poly	1123
9.173.4.2 canvas	1124
9.173.4.3 ctcpnl	1124
9.174TTSupport::TimeRange Class Reference	1124
9.174.1 Detailed Description	1125
9.174.2 Constructor & Destructor Documentation	1125
9.174.2.1 TimeRange() [1/2]	1125
9.174.2.2 TimeRange() [2/2]	1125
9.174.3 Member Function Documentation	1126
9.174.3.1 ContainsTime()	1126

9.174.3.2 From()	1126
9.174.3.3 operator<()	1126
9.174.3.4 operator<=()	1127
9.174.3.5 operator=()	1127
9.174.3.6 operator==(())	1127
9.174.3.7 operator>()	1128
9.174.3.8 operator>=()	1128
9.174.3.9 Read()	1128
9.174.3.10To()	1129
9.174.3.11Write()	1129
9.174.4 Member Data Documentation	1129
9.174.4.1 from	1129
9.174.4.2 to	1129
9.175TTSupport::TimeTableSystem Class Reference	1130
9.175.1 Detailed Description	1133
9.175.2 Constructor & Destructor Documentation	1133
9.175.2.1 TimeTableSystem() [1/3]	1133
9.175.2.2 TimeTableSystem() [2/3]	1133
9.175.2.3 TimeTableSystem() [3/3]	1134
9.175.2.4 ~TimeTableSystem()	1134
9.175.3 Member Function Documentation	1134
9.175.3.1 AddCab()	1134
9.175.3.2 AddNote()	1135
9.175.3.3 AddStation()	1135
9.175.3.4 AddStorageTrack()	1135
9.175.3.5 AddTrain()	1136
9.175.3.6 AddTrainLongVersion()	1136
9.175.3.7 ComputeTimes()	1137



9.175.3.8 CreateLaTeXTimetable()	1138
9.175.3.9 DeleteTrain()	1138
9.175.3.10 DuplicateStationIndex()	1138
9.175.3.11 Filename()	1139
9.175.3.12 FindCab()	1139
9.175.3.13 FindStationByName()	1139
9.175.3.14 FindStorageTrack()	1140
9.175.3.15 FindTrainByName()	1140
9.175.3.16 FindTrainByNumber()	1140
9.175.3.17 FirstCab()	1141
9.175.3.18 FirstPrintOption()	1141
9.175.3.19 FirstTrain()	1141
9.175.3.20 GetPrintOption()	1141
9.175.3.21 IthStation()	1142
9.175.3.22 LastCab()	1142
9.175.3.23 LastPrintOption()	1142
9.175.3.24 LastTrain()	1142
9.175.3.25 MakeTimeTableGroupByClass()	1143
9.175.3.26 MakeTimeTableGroupManually()	1144
9.175.3.27 MakeTimeTableOneTable()	1144
9.175.3.28 MakeTimeTableOneTableStationsCenter()	1146
9.175.3.29 MakeTimeTableOneTableStationsLeft()	1146
9.175.3.30 Name()	1147
9.175.3.31 Note()	1147
9.175.3.32 NumberOfCabs()	1147
9.175.3.33 NumberOfNotes()	1147
9.175.3.34 NumberOfStations()	1148
9.175.3.35 NumberOfTrains()	1148

9.175.3.36ReadNote()	1148
9.175.3.37SetDuplicateStationIndex()	1148
9.175.3.38SetNote()	1149
9.175.3.39SetPrintOption()	1149
9.175.3.40SMile()	1149
9.175.3.41StationName()	1150
9.175.3.42TimeInterval()	1150
9.175.3.43TimeScale()	1150
9.175.3.44TotalLength()	1150
9.175.3.45WriteNewTimeTableFile()	1151
9.175.3.46WriteNote()	1151
9.175.3.47WriteOldTimeTableFile()	1151
9.175.4 Member Data Documentation	1152
9.175.4.1 cabs	1152
9.175.4.2 DirectionName	1152
9.175.4.3 filepath	1152
9.175.4.4 name	1152
9.175.4.5 notes	1153
9.175.4.6 printOptions	1153
9.175.4.7 stations	1153
9.175.4.8 timeinterval	1153
9.175.4.9 timescale	1153
9.175.4.10TOCP	1153
9.175.4.11trains	1154
9.176CTCPanel::Toggle Class Reference	1154
9.176.1 Detailed Description	1155
9.176.2 Constructor & Destructor Documentation	1156
9.176.2.1 Toggle()	1156

9.176.2.2 ~Toggle()	1156
9.176.3 Member Function Documentation	1156
9.176.3.1 _AddTLever()	1156
9.176.3.2 _configureCenterLabel()	1157
9.176.3.3 _configureLeftLabel()	1157
9.176.3.4 _configureRightLabel()	1157
9.176.3.5 _MoveTLever()	1157
9.176.3.6 _VerifyOrientationHV()	1158
9.176.3.7 geti()	1158
9.176.3.8 getv()	1158
9.176.3.9 invoke()	1158
9.176.3.10seti()	1159
9.176.3.11setv()	1159
9.176.4 Member Data Documentation	1160
9.176.4.1 canvas	1160
9.176.4.2 ctcpnl	1160
9.176.4.3 lever	1160
9.177Parsers::TrackBody Class Reference	1160
9.177.1 Detailed Description	1161
9.177.2 Constructor & Destructor Documentation	1161
9.177.2.1 TrackBody()	1161
9.177.2.2 ~TrackBody()	1162
9.177.3 Member Function Documentation	1162
9.177.3.1 AppendTrackBodyElt()	1162
9.177.3.2 ConsTrackBody()	1162
9.177.3.3 TrackBodyLength()	1162
9.177.4 Friends And Related Function Documentation	1162
9.177.4.1 operator<<	1163

9.177.4.2 TrackGraph . . . . .	1163
9.177.5 Member Data Documentation . . . . .	1163
9.177.5.1 element . . . . .	1163
9.177.5.2 next . . . . .	1163
9.178Parsers::TrackBodyElt Class Reference . . . . .	1163
9.178.1 Detailed Description . . . . .	1164
9.178.2 Constructor & Destructor Documentation . . . . .	1164
9.178.2.1 TrackBodyElt() . . . . .	1165
9.178.2.2 ~TrackBodyElt() . . . . .	1165
9.178.3 Member Function Documentation . . . . .	1165
9.178.3.1 ConnectedTrackEnd() . . . . .	1165
9.178.3.2 UnConnectedTrackEnd() . . . . .	1165
9.178.4 Friends And Related Function Documentation . . . . .	1166
9.178.4.1 operator<< . . . . .	1166
9.178.4.2 TrackGraph . . . . .	1166
9.178.5 Member Data Documentation . . . . .	1166
9.178.5.1 a . . . . .	1166
9.178.5.2 index . . . . .	1166
9.178.5.3 x . . . . .	1167
9.178.5.4 y . . . . .	1167
9.179Parsers::TrackGraph Class Reference . . . . .	1167
9.179.1 Detailed Description . . . . .	1172
9.179.2 Member Typedef Documentation . . . . .	1173
9.179.2.1 CompressedEdgePair . . . . .	1173
9.179.2.2 CompressedEdgePairVector . . . . .	1173
9.179.2.3 CompressedGraph . . . . .	1174
9.179.2.4 CompressedIdNodeMap . . . . .	1174
9.179.2.5 CompressedNode . . . . .	1174

9.179.2.6 CompressedNodeVector . . . . .	1174
9.179.2.7 Graph . . . . .	1174
9.179.2.8 IdNodeMap . . . . .	1175
9.179.2.9 Node . . . . .	1175
9.179.3 Member Enumeration Documentation . . . . .	1175
9.179.3.1 NodeType . . . . .	1175
9.179.3.2 RotationUnit . . . . .	1175
9.179.4 Constructor & Destructor Documentation . . . . .	1176
9.179.4.1 TrackGraph() . . . . .	1176
9.179.4.2 ~TrackGraph() . . . . .	1176
9.179.5 Member Function Documentation . . . . .	1176
9.179.5.1 AddNewNode() . . . . .	1176
9.179.5.2 Angle() . . . . .	1176
9.179.5.3 compressed_edge_exists() . . . . .	1177
9.179.5.4 CompressedEdgeCount() . . . . .	1177
9.179.5.5 CompressedEdgeLength() . . . . .	1177
9.179.5.6 CompressedEdgeNode() . . . . .	1177
9.179.5.7 CompressedGraphCircleLayout() . . . . .	1178
9.179.5.8 CompressedGraphKamadaKawaiSpring() . . . . .	1178
9.179.5.9 CompressedGraphKruskalMinimumSpanningTree() . . . . .	1178
9.179.5.10CompressedGraphPrimMinimumSpanningTree() . . . . .	1178
9.179.5.11CompressedNodePositionX() . . . . .	1179
9.179.5.12CompressedNodePositionY() . . . . .	1179
9.179.5.13CompressedNodeSegments() . . . . .	1179
9.179.5.14CompressGraph() . . . . .	1179
9.179.5.15computeHeads() . . . . .	1179
9.179.5.16ComputeRouteLength() . . . . .	1180
9.179.5.17DeleteTurnoutGraphic() . . . . .	1180

9.179.5.18DeleteTurnoutRouteList()	1180
9.179.5.19EdgeA()	1180
9.179.5.20EdgeIndex()	1180
9.179.5.21EdgeLength()	1181
9.179.5.22EdgeX()	1181
9.179.5.23EdgeY()	1181
9.179.5.24FindBlock()	1181
9.179.5.25FindNode()	1182
9.179.5.26Heads()	1182
9.179.5.27HighestNode()	1182
9.179.5.28InsertBezierTrack()	1182
9.179.5.29InsertBlock()	1183
9.179.5.30InsertCompressedNode()	1183
9.179.5.31InsertControl()	1183
9.179.5.32InsertCornuTrack()	1183
9.179.5.33InsertCurveTrack()	1184
9.179.5.34InsertJointTrack()	1184
9.179.5.35InsertSensor()	1184
9.179.5.36InsertSignal()	1184
9.179.5.37InsertStraightTrack()	1185
9.179.5.38InsertSwitchMotor()	1185
9.179.5.39InsertTurnOut()	1185
9.179.5.40InsertTurnTable()	1185
9.179.5.41IsCompressed()	1186
9.179.5.42IsCompressedNode()	1186
9.179.5.43IsNodeP()	1186
9.179.5.44IsNone()	1186
9.179.5.45LengthOfCurve()	1187

9.179.5.46LengthOfJoint()	1187
9.179.5.47LengthOfNode()	1187
9.179.5.48LengthOfStraight()	1187
9.179.5.49LowestNode()	1188
9.179.5.50MakeTurnoutGraphic()	1188
9.179.5.51MakeTurnoutRouteList()	1188
9.179.5.52NameOfNode()	1188
9.179.5.53NodeTurnoutGraphic()	1189
9.179.5.54NodeTurnoutRoutelist()	1189
9.179.5.55NormalActionScript()	1189
9.179.5.56NumberOfHeads()	1189
9.179.5.57NumEdges()	1189
9.179.5.58OffScript()	1189
9.179.5.59OnScript()	1190
9.179.5.60OrigX()	1190
9.179.5.61OrigY()	1190
9.179.5.62ReverseActionScript()	1191
9.179.5.63Roots()	1191
9.179.5.64SenseScript()	1191
9.179.5.65SignalAspects()	1191
9.179.5.66r_rotate()	1192
9.179.5.67r_scale() [1/2]	1192
9.179.5.68r_scale() [2/2]	1192
9.179.5.69r_translate()	1192
9.179.5.70TrackList()	1192
9.179.5.71traversePrimMST()	1193
9.179.5.72TurnoutNumber()	1193
9.179.5.73TypeOfNode()	1193

9.179.6 Friends And Related Function Documentation . . . . .	1193
9.179.6.1 operator<< . . . . .	1193
9.179.7 Member Data Documentation . . . . .	1193
9.179.7.1 backpointers . . . . .	1194
9.179.7.2 c_idMap . . . . .	1194
9.179.7.3 c_nodes . . . . .	1194
9.179.7.4 c_roots . . . . .	1194
9.179.7.5 circleLayoutP . . . . .	1194
9.179.7.6 compressedP . . . . .	1194
9.179.7.7 heads . . . . .	1195
9.179.7.8 idMap . . . . .	1195
9.179.7.9 KamadaKawaiSpringLayoutP . . . . .	1195
9.179.7.10nodes . . . . .	1195
9.179.7.11none . . . . .	1195
9.179.7.12valid_heads . . . . .	1195
9.180TTSupport::Train Class Reference . . . . .	1196
9.180.1 Detailed Description . . . . .	1197
9.180.2 Constructor & Destructor Documentation . . . . .	1197
9.180.2.1 Train() . . . . .	1198
9.180.3 Member Function Documentation . . . . .	1198
9.180.3.1 AddNoteToStop() . . . . .	1198
9.180.3.2 AddNoteToTrain() . . . . .	1199
9.180.3.3 ClassNumber() . . . . .	1199
9.180.3.4 Departure() . . . . .	1199
9.180.3.5 Name() . . . . .	1199
9.180.3.6 Note() . . . . .	1199
9.180.3.7 Number() . . . . .	1200
9.180.3.8 NumberOfNotes() . . . . .	1200



9.180.3.9 NumberOfStops()	1200
9.180.3.10Read()	1200
9.180.3.11RemoveNoteFromStop()	1201
9.180.3.12RemoveNoteFromTrain()	1201
9.180.3.13SetDeparture()	1201
9.180.3.14SetDestinationStorageTrack()	1202
9.180.3.15SetOriginStorageTrack()	1202
9.180.3.16SetTransitStorageTrack()	1202
9.180.3.17Speed()	1202
9.180.3.18StartSMile()	1203
9.180.3.19StopI()	1203
9.180.3.20UpdateStopCab()	1203
9.180.3.21UpdateStopLayover()	1204
9.180.3.22Write()	1204
9.180.4 Member Data Documentation	1204
9.180.4.1 classnumber	1204
9.180.4.2 departure	1204
9.180.4.3 name	1205
9.180.4.4 notes	1205
9.180.4.5 number	1205
9.180.4.6 speed	1205
9.180.4.7 startSMile	1205
9.180.4.8 stops	1205
9.181FCFSupport::Train Class Reference	1206
9.181.1 Detailed Description	1208
9.181.2 Member Enumeration Documentation	1208
9.181.2.1 TrainType	1208
9.181.3 Constructor & Destructor Documentation	1209

9.181.3.1 Train() [1/3]	1209
9.181.3.2 Train() [2/3]	1209
9.181.3.3 Train() [3/3]	1210
9.181.3.4 ~Train()	1210
9.181.4 Member Function Documentation	1211
9.181.4.1 CarTypes()	1211
9.181.4.2 Description()	1211
9.181.4.3 DivisionList()	1211
9.181.4.4 Done()	1211
9.181.4.5 IndustryStop()	1211
9.181.4.6 MaxCars()	1212
9.181.4.7 MaxClear()	1212
9.181.4.8 MaxLength()	1212
9.181.4.9 MaxWeight()	1212
9.181.4.10Name()	1213
9.181.4.11NumberOfOrders()	1213
9.181.4.12NumberOfStops()	1213
9.181.4.13OnDuty()	1213
9.181.4.14operator=()	1213
9.181.4.15Order()	1214
9.181.4.16Print()	1214
9.181.4.17SetMaxLength()	1214
9.181.4.18SetMaxWeight()	1215
9.181.4.19SetPrint()	1215
9.181.4.20SetShift()	1215
9.181.4.21Shift()	1216
9.181.4.22StationStop()	1216
9.181.4.23Type()	1216

9.181.5 Friends And Related Function Documentation . . . . .	1216
9.181.5.1 System . . . . .	1217
9.181.6 Member Data Documentation . . . . .	1217
9.181.6.1 carTypes . . . . .	1217
9.181.6.2 description . . . . .	1217
9.181.6.3 divList . . . . .	1217
9.181.6.4 done . . . . .	1217
9.181.6.5 maxcars . . . . .	1218
9.181.6.6 maxclear . . . . .	1218
9.181.6.7 maxlength . . . . .	1218
9.181.6.8 maxweight . . . . .	1218
9.181.6.9 name . . . . .	1218
9.181.6.10 onduy . . . . .	1219
9.181.6.11 orders . . . . .	1219
9.181.6.12 print . . . . .	1219
9.181.6.13 shift . . . . .	1219
9.181.6.14 stops . . . . .	1219
9.181.6.15 type . . . . .	1220
9.182 FCFSupport::TrainDisplayCallback Class Reference . . . . .	1220
9.182.1 Detailed Description . . . . .	1220
9.182.2 Constructor & Destructor Documentation . . . . .	1220
9.182.2.1 TrainDisplayCallback() . . . . .	1221
9.182.2.2 ~TrainDisplayCallback() . . . . .	1221
9.182.3 Member Function Documentation . . . . .	1221
9.182.3.1 CloseTrainDisplay() . . . . .	1221
9.182.3.2 GrabTrainDisplay() . . . . .	1221
9.182.3.3 InitializeTrainDisplay() . . . . .	1221
9.182.3.4 ReleaseTrainDisplay() . . . . .	1222

9.182.3.5 UpdateTrainDisplay()	1222
9.183Parsers::TrackGraph::Transform2D Class Reference	1223
9.183.1 Detailed Description	1224
9.183.2 Constructor & Destructor Documentation	1224
9.183.2.1 Transform2D() [1/3]	1224
9.183.2.2 Transform2D() [2/3]	1224
9.183.2.3 Transform2D() [3/3]	1224
9.183.3 Member Function Documentation	1225
9.183.3.1 Apply() [1/2]	1225
9.183.3.2 Apply() [2/2]	1225
9.183.3.3 Determinant()	1225
9.183.3.4 Inverse()	1225
9.183.3.5 Minor()	1226
9.183.3.6 operator"!="()	1226
9.183.3.7 operator==(())	1226
9.183.4 Friends And Related Function Documentation	1226
9.183.4.1 operator*	1226
9.183.5 Member Data Documentation	1226
9.183.5.1 FUZZ	1227
9.183.5.2 matrix	1227
9.184Parsers::TurnoutBody Class Reference	1227
9.184.1 Detailed Description	1228
9.184.2 Constructor & Destructor Documentation	1228
9.184.2.1 TurnoutBody()	1228
9.184.3 Member Function Documentation	1228
9.184.3.1 CleanUpElement()	1228
9.184.3.2 CleanUpTurnoutBody()	1229
9.184.3.3 ConsTurnoutBody()	1229

9.184.3.4 Element()	1229
9.184.3.5 TurnoutEnds()	1229
9.184.3.6 TurnoutRouteCount()	1229
9.184.3.7 TurnoutSegmentCount()	1230
9.184.4 Friends And Related Function Documentation	1230
9.184.4.1 TrackGraph	1230
9.184.4.2 TurnoutBodyElt	1230
9.184.5 Member Data Documentation	1230
9.184.5.1 element	1230
9.184.5.2 next	1230
9.185Parsers::TurnoutBodyElt Class Reference	1231
9.185.1 Detailed Description	1232
9.185.2 Member Enumeration Documentation	1233
9.185.2.1 TurnoutBodyEltType	1233
9.185.3 Constructor & Destructor Documentation	1233
9.185.3.1 TurnoutBodyElt()	1233
9.185.3.2 ~TurnoutBodyElt()	1233
9.185.4 Member Function Documentation	1233
9.185.4.1 GetTurnoutCurveSegment()	1234
9.185.4.2 GetTurnoutJointSegment()	1234
9.185.4.3 GetTurnoutRoute()	1234
9.185.4.4 GetTurnoutStraightSegment()	1234
9.185.4.5 InitTSegId()	1235
9.185.4.6 MakeTurnoutCurveSegment()	1235
9.185.4.7 MakeTurnoutEnd()	1235
9.185.4.8 MakeTurnoutJointSegment()	1235
9.185.4.9 MakeTurnoutRoute()	1236
9.185.4.10MakeTurnoutStraightSegment()	1236

9.185.4.11TheType()	1236
9.185.5 Friends And Related Function Documentation	1236
9.185.5.1 TrackGraph	1236
9.185.5.2 TurnoutBody	1236
9.185.6 Member Data Documentation	1237
9.185.6.1 ang0	1237
9.185.6.2 ang1	1237
9.185.6.3 L	1237
9.185.6.4 pos1	1237
9.185.6.5 pos2	1238
9.185.6.6 R	1238
9.185.6.7 radius	1238
9.185.6.8 routeList	1238
9.185.6.9 RouteName	1238
9.185.6.10segCount	1239
9.185.6.11segmentId	1239
9.185.6.12theEnd	1239
9.185.6.13theType	1239
9.186Parsers::TurnoutGraphic Struct Reference	1239
9.186.1 Detailed Description	1240
9.186.2 Member Data Documentation	1240
9.186.2.1 maxX	1240
9.186.2.2 maxY	1240
9.186.2.3 minX	1241
9.186.2.4 minY	1241
9.186.2.5 numSegments	1241
9.186.2.6 segments	1241
9.187Parsers::TurnoutRoutelist Struct Reference	1241

9.187.1 Detailed Description . . . . .	1242
9.187.2 Member Data Documentation . . . . .	1242
9.187.2.1 numRoutelists . . . . .	1242
9.187.2.2 routes . . . . .	1242
9.188FCFSupport::PDFFileStructures::Type1FontDictionary Class Reference . . . . .	1242
9.188.1 Detailed Description . . . . .	1243
9.188.2 Constructor & Destructor Documentation . . . . .	1243
9.188.2.1 Type1FontDictionary() [1/2] . . . . .	1244
9.188.2.2 Type1FontDictionary() [2/2] . . . . .	1244
9.188.2.3 ~Type1FontDictionary() . . . . .	1245
9.188.3 Member Function Documentation . . . . .	1245
9.188.3.1 WriteDictionaryElements() . . . . .	1245
9.188.4 Member Data Documentation . . . . .	1245
9.188.4.1 baseFont . . . . .	1246
9.188.4.2 encodingDictionary . . . . .	1246
9.188.4.3 encodingName . . . . .	1246
9.188.4.4 firstChar . . . . .	1246
9.188.4.5 fontDescriptor . . . . .	1246
9.188.4.6 lastChar . . . . .	1246
9.188.4.7 widths . . . . .	1247
9.189FCFSupport::PDFFileStructures::TypedDictionary Class Reference . . . . .	1247
9.189.1 Detailed Description . . . . .	1248
9.189.2 Constructor & Destructor Documentation . . . . .	1248
9.189.2.1 TypedDictionary() . . . . .	1248
9.189.2.2 ~TypedDictionary() . . . . .	1248
9.189.3 Member Function Documentation . . . . .	1248
9.189.3.1 WriteDictionaryElements() . . . . .	1249
9.189.3.2 WriteDictionaryType() . . . . .	1249

9.189.4 Member Data Documentation . . . . .	1249
9.189.4.1 type . . . . .	1249
9.190FCFSupport::WorkInProgressCallback Class Reference . . . . .	1250
9.190.1 Detailed Description . . . . .	1250
9.190.2 Constructor & Destructor Documentation . . . . .	1250
9.190.2.1 WorkInProgressCallback() . . . . .	1250
9.190.2.2 ~WorkInProgressCallback() . . . . .	1251
9.190.3 Member Function Documentation . . . . .	1251
9.190.3.1 ProgressDone() . . . . .	1251
9.190.3.2 ProgressStart() . . . . .	1251
9.190.3.3 ProgressUpdate() . . . . .	1252
9.191xpressnet::XPressNet Class Reference . . . . .	1252
9.191.1 Detailed Description . . . . .	1255
9.191.2 Constructor & Destructor Documentation . . . . .	1255
9.191.2.1 XPressNet() . . . . .	1255
9.191.2.2 ~XPressNet() . . . . .	1256
9.191.3 Member Function Documentation . . . . .	1256
9.191.3.1 _ADDRESS() . . . . .	1256
9.191.3.2 _appendXORByte() . . . . .	1256
9.191.3.3 _CheckForResponse_0x00() . . . . .	1256
9.191.3.4 _CheckForResponse_0x40() . . . . .	1257
9.191.3.5 _CheckForResponse_0x60() . . . . .	1257
9.191.3.6 _CheckForResponse_0x80() . . . . .	1257
9.191.3.7 _CheckForResponse_0xa0() . . . . .	1257
9.191.3.8 _CheckForResponse_0xc0() . . . . .	1258
9.191.3.9 _CheckForResponse_0xe0() . . . . .	1258
9.191.3.10_readbyte() . . . . .	1258
9.191.3.11_readevent() . . . . .	1258



9.191.3.12_timeoutevent()	1259
9.191.3.13_transmit()	1259
9.191.3.14AccessoryDecoderInformationRequest()	1259
9.191.3.15AccessoryDecoderOperation()	1259
9.191.3.16AddLocomotiveToMultiUnit()	1260
9.191.3.17AddressInquiryNextMU()	1260
9.191.3.18AddressInquiryNextMUMember()	1260
9.191.3.19AddressInquiryNextStack()	1261
9.191.3.20AddressInquiryPreviousMU()	1261
9.191.3.21AddressInquiryPreviousMUMember()	1261
9.191.3.22AddressInquiryPreviousStack()	1262
9.191.3.23CheckForResponse()	1262
9.191.3.24CommandStationSoftwareVersion()	1262
9.191.3.25CommandStationStatusRequest()	1262
9.191.3.26DeleteLocomotiveFromStack()	1262
9.191.3.27DirectModeCVRead()	1263
9.191.3.28DirectModeCVWrite()	1263
9.191.3.29DissolveDoubleHeader()	1263
9.191.3.30EmergencyStopAllLocomotives()	1264
9.191.3.31EmergencyStopALocomotive()	1264
9.191.3.32EstablishDoubleHeader()	1264
9.191.3.33FunctionStatusRequest()	1264
9.191.3.34GetLI100VersionNumbers()	1265
9.191.3.35GetNextCommandStationResponse()	1265
9.191.3.36LocomotiveInformationRequest()	1265
9.191.3.37OperatingModeProgrammingBitModeWrite()	1265
9.191.3.38OperatingModeProgrammingByteModeWrite()	1266
9.191.3.39PagedModeCVRead()	1266

9.191.3.40	PagedModeCVWrite()	1267
9.191.3.41	readevent()	1267
9.191.3.42	RegisterModeRead()	1267
9.191.3.43	RegisterModeWrite()	1267
9.191.3.44	RemoveLocomotiveFromMultiUnit()	1268
9.191.3.45	RequestForServiceModeResults()	1268
9.191.3.46	ResumeOperations()	1268
9.191.3.47	SetCommandStationPowerUpMode()	1268
9.191.3.48	SetFunctionStateGroup1()	1269
9.191.3.49	SetFunctionStateGroup2()	1269
9.191.3.50	SetFunctionStateGroup3()	1270
9.191.3.51	SetLI101Address()	1270
9.191.3.52	SetLocomotiveFunctionsGroup1()	1270
9.191.3.53	SetLocomotiveFunctionsGroup2()	1271
9.191.3.54	SetLocomotiveFunctionsGroup3()	1271
9.191.3.55	SetLocomotiveSpeedAndDirection()	1271
9.191.3.56	StopOperations()	1272
9.191.4	Member Data Documentation	1272
9.191.4.1	_timeout	1272
9.191.4.2	responseList	1272
9.191.4.3	ttyfd	1273
9.192	xpressnet::XpressNetEvent Class Reference	1273
9.192.1	Detailed Description	1273
9.192.2	Constructor & Destructor Documentation	1273
9.192.2.1	XpressNetEvent()	1274
9.192.2.2	~XpressNetEvent()	1274
9.192.3	Member Function Documentation	1274
9.192.3.1	_eventhandler()	1274

9.192.4 Member Data Documentation . . . . .	1274
9.192.4.1 _script . . . . .	1275
9.192.4.2 xpressnet . . . . .	1275
9.193YY_MRRXtrkCad_INHERIT Class Reference . . . . .	1275
9.193.1 Member Enumeration Documentation . . . . .	1276
9.193.1.1 YY_MRRXtrkCad_ENUM_TOKEN . . . . .	1276
9.193.2 Constructor & Destructor Documentation . . . . .	1278
9.193.2.1 ~MRRXtrkCad() . . . . .	1278
9.193.3 Member Function Documentation . . . . .	1278
9.193.3.1 lookup_word() . . . . .	1278
9.193.3.2 YY_MRRXtrkCad_CLASS() . . . . .	1278
9.193.3.3 YY_MRRXtrkCad_ERROR() . . . . .	1278
9.193.3.4 YY_MRRXtrkCad_LEX() . . . . .	1279
9.193.3.5 YY_MRRXtrkCad_PARSE() . . . . .	1279
9.193.3.6 yyerror1() . . . . .	1279
9.193.4 Member Data Documentation . . . . .	1279
9.193.4.1 CurrentScale . . . . .	1279
9.193.4.2 fieldflag . . . . .	1279
9.193.4.3 scanEol . . . . .	1279
9.193.4.4 scanToEND . . . . .	1280
9.193.4.5 YY_MRRXtrkCad_CHAR . . . . .	1280
9.193.4.6 YY_MRRXtrkCad_DEBUG_FLAG . . . . .	1280
9.193.4.7 YY_MRRXtrkCad_LLOC . . . . .	1280
9.193.4.8 YY_MRRXtrkCad_LVAL . . . . .	1280
9.193.4.9 YY_MRRXtrkCad_NERRS . . . . .	1280
9.194yy_MRRXtrkCad_stype Union Reference . . . . .	1280
9.194.1 Member Data Documentation . . . . .	1281
9.194.1.1 fval . . . . .	1281

9.194.1.2 il	1281
9.194.1.3 ival	1281
9.194.1.4 spl	1281
9.194.1.5 sval	1282
9.194.1.6 tb	1282
9.194.1.7 tbb	1282
9.194.1.8 tbbe	1282
9.194.1.9 tbe	1282
9.194.1.10 tcb	1282
9.194.1.11 tcbe	1282
9.194.1.12 rb	1283
9.194.1.13 rbe	1283
9.195yytype Struct Reference	1283
9.195.1 Member Data Documentation	1283
9.195.1.1 first_column	1283
9.195.1.2 first_line	1283
9.195.1.3 last_column	1284
9.195.1.4 last_line	1284
9.195.1.5 text	1284
9.195.1.6 timestamp	1284

<b>10 File Documentation</b>	<b>1285</b>
10.1 C++/Azatrax/Azatrax.h File Reference	1285
10.1.1 Macro Definition Documentation	1285
10.1.1.1 ErrorCode	1286
10.1.1.2 stopwatchFract	1286
10.1.1.3 stopwatchSeconds	1286
10.2 C++/Azatrax/mrd.h File Reference	1286
10.3 C++/Azatrax/sl2.h File Reference	1286
10.4 C++/Azatrax/sr4.h File Reference	1287
10.5 C++/FCFSupport/CallBack.h File Reference	1287
10.6 C++/FCFSupport/Car.h File Reference	1288
10.7 C++/FCFSupport/CarType.h File Reference	1288
10.8 C++/FCFSupport/Division.h File Reference	1289
10.9 C++/FCFSupport/FCFSupportGroup.h File Reference	1290
10.10C++/FCFSupport/Industry.h File Reference	1290
10.11C++/FCFSupport/LQ24Printer.h File Reference	1290
10.12C++/FCFSupport/Owner.h File Reference	1291
10.13C++/FCFSupport/PathName.h File Reference	1291
10.14C++/TTSupport/PathName.h File Reference	1292
10.15C++/FCFSupport/PDFPrinter.h File Reference	1292
10.15.1 Macro Definition Documentation	1293
10.15.1.1 oneColumnWidthFraction	1293
10.16C++/FCFSupport/PDFPrinterSupport.h File Reference	1293
10.16.1 Function Documentation	1295
10.16.1.1 asctime_r()	1295
10.16.1.2 localtime_r()	1295
10.17C++/FCFSupport/PostScriptPrinter.h File Reference	1295
10.18C++/FCFSupport/Printer.h File Reference	1296

10.19C++/FCFSupport/Station.h File Reference . . . . .	1296
10.20C++/TTSupport/Station.h File Reference . . . . .	1297
10.21C++/FCFSupport/SwitchList.h File Reference . . . . .	1297
10.22C++/FCFSupport/System.h File Reference . . . . .	1298
10.23C++/FCFSupport/TextPrinter.h File Reference . . . . .	1299
10.24C++/FCFSupport/Train.h File Reference . . . . .	1299
10.25C++/TTSupport/Train.h File Reference . . . . .	1300
10.26C++/ParserClasses/BezierBody.h File Reference . . . . .	1300
10.26.1 Macro Definition Documentation . . . . .	1301
10.26.1.1 angle . . . . .	1301
10.26.1.2 len0 . . . . .	1301
10.26.1.3 len1 . . . . .	1301
10.27C++/ParserClasses/CornuBody.h File Reference . . . . .	1302
10.27.1 Macro Definition Documentation . . . . .	1302
10.27.1.1 angle . . . . .	1302
10.27.1.2 len0 . . . . .	1302
10.27.1.3 len1 . . . . .	1303
10.28C++/ParserClasses/IntegerList.h File Reference . . . . .	1303
10.29C++/ParserClasses/MRRXtrkCad.tab.h File Reference . . . . .	1303
10.29.1 Macro Definition Documentation . . . . .	1305
10.29.1.1 BISON_YYLTYPE_ISDECLARED . . . . .	1305
10.29.1.2 FALSE . . . . .	1305
10.29.1.3 RADIANS . . . . .	1305
10.29.1.4 TRUE . . . . .	1305
10.29.1.5 YY_MRRXtrkCad_CHAR . . . . .	1305
10.29.1.6 YY_MRRXtrkCad_CLASS . . . . .	1305
10.29.1.7 YY_MRRXtrkCad_COMPATIBILITY . . . . .	1306
10.29.1.8 YY_MRRXtrkCad_CONSTRUCTOR_CODE . . . . .	1306

10.29.1.9 YY_MRRXtrkCad_CONSTRUCTOR_INIT . . . . .	1306
10.29.1.10YY_MRRXtrkCad_CONSTRUCTOR_PARAM . . . . .	1306
10.29.1.11YY_MRRXtrkCad_DEBUG . . . . .	1306
10.29.1.12YY_MRRXtrkCad_DEBUG_FLAG . . . . .	1306
10.29.1.13YY_MRRXtrkCad_ERROR . . . . .	1307
10.29.1.14YY_MRRXtrkCad_ERROR_BODY . . . . .	1307
10.29.1.15YY_MRRXtrkCad_ERROR_VERBOSE . . . . .	1307
10.29.1.16YY_MRRXtrkCad_INHERIT . . . . .	1307
10.29.1.17YY_MRRXtrkCad_LEX . . . . .	1307
10.29.1.18YY_MRRXtrkCad_LEX_BODY . . . . .	1307
10.29.1.19YY_MRRXtrkCad_LLOC . . . . .	1307
10.29.1.20YY_MRRXtrkCad_LSP_NEEDED . . . . .	1308
10.29.1.21YY_MRRXtrkCad_LTYPE . . . . .	1308
10.29.1.22YY_MRRXtrkCad_LVAL . . . . .	1308
10.29.1.23YY_MRRXtrkCad_MEMBERS . . . . .	1308
10.29.1.24YY_MRRXtrkCad_NERRS . . . . .	1308
10.29.1.25YY_MRRXtrkCad_PARSE . . . . .	1308
10.29.1.26YY_MRRXtrkCad_PARSE_PARAM . . . . .	1309
10.29.1.27YY_MRRXtrkCad_STYPE . . . . .	1309
10.29.1.28YY_MRRXtrkCad_USE_CONST_TOKEN . . . . .	1309
10.29.1.29YY_MRRXtrkCad_USE_GOTO . . . . .	1309
10.29.1.30YY_USE_CLASS . . . . .	1309
10.29.2 Typedef Documentation . . . . .	1309
10.29.2.1 yyltype . . . . .	1309
10.29.3 Variable Documentation . . . . .	1309
10.29.3.1 CENTIMETERSperMM . . . . .	1310
10.29.3.2 FEETperMM . . . . .	1310
10.29.3.3 GScale . . . . .	1310

10.29.3.4 HOScale . . . . .	1310
10.29.3.5 INCHESperMM . . . . .	1310
10.29.3.6 IScale . . . . .	1310
10.29.3.7 METERSperMM . . . . .	1310
10.29.3.8 NScale . . . . .	1311
10.29.3.9 OScale . . . . .	1311
10.29.3.10YARDSperMM . . . . .	1311
10.30C++/ParserClasses/ParseFile.h File Reference . . . . .	1311
10.31C++/ParserClasses/ParserClassesGroup.h File Reference . . . . .	1311
10.32C++/ParserClasses/SocketPair.h File Reference . . . . .	1312
10.33C++/ParserClasses/TrackBody.h File Reference . . . . .	1312
10.34C++/ParserClasses/TrackGraph.h File Reference . . . . .	1312
10.35C++/ParserClasses/TurnoutBody.h File Reference . . . . .	1313
10.35.1 Macro Definition Documentation . . . . .	1314
10.35.1.1 angle . . . . .	1314
10.35.1.2 len0 . . . . .	1314
10.35.1.3 len1 . . . . .	1314
10.36C++/RailDriver/RaildriverIO.h File Reference . . . . .	1314
10.37C++/TclSocketCAN/TclSocketCAN.i File Reference . . . . .	1315
10.37.1 Macro Definition Documentation . . . . .	1315
10.37.1.1 SWIG_name . . . . .	1315
10.37.1.2 SWIG_version . . . . .	1315
10.38C++/TTSupport/Cab.h File Reference . . . . .	1315
10.39C++/TTSupport/TimeTableSystem.h File Reference . . . . .	1316
10.39.1 Macro Definition Documentation . . . . .	1317
10.39.1.1 USE_UNORDERED_MAP . . . . .	1317
10.40C++/TTSupport/TimeTableSystemTcl.h File Reference . . . . .	1318
10.41C++/TTSupport/TTSupportGroup.h File Reference . . . . .	1318



10.42C++/wiringPi/tclwiringpi.i File Reference . . . . .	1318
10.42.1 Macro Definition Documentation . . . . .	1319
10.42.1.1 SWIG_name . . . . .	1319
10.42.1.2 SWIG_version . . . . .	1319
10.42.2 Variable Documentation . . . . .	1319
10.42.2.1 Tclwiringpi . . . . .	1319
10.43Doc/doxygen/titlepage.h File Reference . . . . .	1320
10.44Scripts/CMri/cmri.tcl File Reference . . . . .	1320
10.45Scripts/Common/CabWidgets.tcl File Reference . . . . .	1320
10.46Scripts/Common/CommonTclGroup.h File Reference . . . . .	1321
10.47Scripts/Common/CTCPanel2.tcl File Reference . . . . .	1321
10.48Scripts/Common/fileentry.tcl File Reference . . . . .	1322
10.49Scripts/Common/gettext.tcl File Reference . . . . .	1323
10.50Scripts/Common/HTMLHelp.tcl File Reference . . . . .	1323
10.51Scripts/Common/labelcombobox.tcl File Reference . . . . .	1324
10.52Scripts/Common/labelselectcolor.tcl File Reference . . . . .	1325
10.53Scripts/Common/labelspinbox.tcl File Reference . . . . .	1325
10.54Scripts/Common/mainwindow.tcl File Reference . . . . .	1326
10.55Scripts/Common/panedw.tcl File Reference . . . . .	1326
10.56Scripts/Common/ParseXML.tcl File Reference . . . . .	1326
10.56.1 Detailed Description . . . . .	1326
10.57Scripts/Common/ReadConfiguration.tcl File Reference . . . . .	1326
10.58Scripts/Common/snitScrollNotebook.tcl File Reference . . . . .	1327
10.59Scripts/Common/splash.tcl File Reference . . . . .	1327
10.60Scripts/ControlSupport/CmriSupport.tcl File Reference . . . . .	1327
10.61Scripts/CTIAcela/CTIAcela.tcl File Reference . . . . .	1328
10.62Scripts/GRSupport/grsupport2.tcl File Reference . . . . .	1328
10.63Scripts/GRSupport/GRSupportTclGroup.h File Reference . . . . .	1329
10.64Scripts/GRSupport/Instruments2.tcl File Reference . . . . .	1329
10.65Scripts/GRSupport/LCARSWidgets2.tcl File Reference . . . . .	1330
10.66Scripts/GRSupport/OvalWidgets2.tcl File Reference . . . . .	1330
10.67Scripts/LCC/ConfigDialogs.tcl File Reference . . . . .	1331
10.68Scripts/LCC/ConfigurationEditor.tcl File Reference . . . . .	1332
10.69Scripts/LCC/eventDialogs.tcl File Reference . . . . .	1332
10.70Scripts/LCC/lcc.tcl File Reference . . . . .	1333
10.71Scripts/LinuxGpio/LinuxGpio.tcl File Reference . . . . .	1335
10.72Scripts/NCE/nce.tcl File Reference . . . . .	1336
10.73Scripts/RailDriverSupport/raildriver_client.tcl File Reference . . . . .	1337
10.74Scripts/Satellite/Satellite.tcl File Reference . . . . .	1338
10.75Scripts/XPressNet/xpressnet.tcl File Reference . . . . .	1338

[Index](#)

1341

# Chapter 1

## Internals (developer) documentation.

### Author

Robert Heller  
Deepwoods Software  
Wendell, MA, USA  
<http://www.deepsoft.com/>  
heller@deepsoft.com

### 1.1 Introduction

This is the documentation for all of the library functions, classes, and widgets. This includes both the C++ implemented libraries and the Tcl implemented libraries. For most of the C++ implemented code there is a Tcl interface included as part of the library. The code is broken up into modules and almost everything is in either a C++ or Tcl namespace. This allows for multiple code libraries to be included or linked to without global name conflicts.

Note: The Tcl code is documented using a C/C++ style. This might seem a little weird from a Tcl point of view.



## Chapter 2

# Module Index

### 2.1 Modules

Here is a list of all modules:

Azatrax . . . . .	23
FCFSupportModule . . . . .	24
ParserClasses . . . . .	25
TimeTableSystem . . . . .	26
TimeTableSystemTcl . . . . .	27
Station . . . . .	34
Train and support classes. . . . .	36
Cab . . . . .	37
TclSocketCANModule . . . . .	38
TclwiringpiModule . . . . .	40
Cmri . . . . .	41
CTIAcela . . . . .	42
LCCModule . . . . .	63
XPressNetModule . . . . .	64
NCEModule . . . . .	65
TclCommon . . . . .	66
LinuxGpio . . . . .	67
Cmri Support code . . . . .	68
RaildriverClientModule . . . . .	69
Graphics Support Code . . . . .	70
GRSupportModule . . . . .	71
SatelliteModule . . . . .	72
FCFSupport . . . . .	73



## Chapter 3

# Namespace Index

### 3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">azatrax</a>		
<a href="#">Azatrax</a>	C++ LibUSB 1.0 Interface . . . . .	75
<a href="#">CabWidgets</a>		
	Cab Widget code . . . . .	76
<a href="#">cmri</a>		
	CMR/I Tcl Serial Port Interface . . . . .	76
<a href="#">CmriSupport</a>		
	Cmri Support code . . . . .	79
<a href="#">CTCPanel</a>		
	CTC Panel code, Version 2 . . . . .	79
<a href="#">ctiacela</a>		
	CTI Acela Tcl Serial Port Interface . . . . .	83
<a href="#">FCFSupport</a>		
	Namespace to hold all of the FCF Support code . . . . .	84
<a href="#">FCFSupport::PDFFileStructures</a>		
	PDF File support structures . . . . .	90
<a href="#">FileEntry</a>		
	This is a specialized form of the LabelEntry widget intended for selecting file names . . . . .	93
<a href="#">gettext</a>		
	Localization functions . . . . .	97
<a href="#">GRSupport</a>		
	Code to support the various graphics packages . . . . .	98
<a href="#">HTMLHelp</a>		
	<a href="#">HTMLHelp</a> namespace, which contains the <a href="#">HTMLHelp</a> snit widget adapter object and associated code . . . . .	102
<a href="#">Instruments</a>		
	Namespace used for instruments code . . . . .	103
<a href="#">LabelComboBox</a>		
	This is a specialized form of the LabelFrame widget containing a ComboBox Widget . . . . .	104
<a href="#">LabelSelectColor</a>		
	This package provides a BWidget style megawidget for selecting colors, in the same style as a LabelEntry widget . . . . .	110

<a href="#">LabelSpinBox</a>	This is a specialized form of the LabelFrame widget containing a SpinBox Widget . . . . .	114
<a href="#">LCARS</a>	Namespace where the <a href="#">LCARS</a> code lives . . . . .	119
<a href="#">lcc</a>	Namespace that holds the LCC interface code . . . . .	119
<a href="#">linuxgpio</a>	Linux GPIO Interface . . . . .	126
<a href="#">nce</a>	Namespace that holds the <a href="#">NCE</a> interface code . . . . .	128
<a href="#">OvalWidgets</a>	These oval shaped widgets are much like the Star Trek NG computer screens . . . . .	135
<a href="#">PanedWindow</a>	A modified version of the BWidget <a href="#">PanedWindow</a> . . . . .	141
<a href="#">Parsers</a>	File-based parser classes . . . . .	141
<a href="#">raildriver</a>	Namespace that holds the Raildriver Client class code . . . . .	143
<a href="#">ReadConfiguration</a>	The Read Configuration File code is contained in this namespace . . . . .	145
<a href="#">TTSupport</a>	Time Table Support Namespace . . . . .	148
<a href="#">xpressnet</a>	Namespace that holds the <a href="#">XPressNet</a> interface code . . . . .	152



## Chapter 4

# Hierarchical Index

### 4.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

xpressnet::AccessoryDecoderInformation . . . . .	161
Instruments::AnalogClock . . . . .	166
azatrax::Azatrax . . . . .	168
azatrax::MRD . . . . .	621
azatrax::SL2 . . . . .	913
azatrax::SR4 . . . . .	928
Parsers::BezierBody . . . . .	182
Parsers::BezierBodyElt . . . . .	185
TTSupport::Cab . . . . .	192
Instruments::CabSignalLamp . . . . .	196
lcc::CanAlias . . . . .	200
lcc::CANGridConnect . . . . .	203
lcc::CANGridConnectOverCANSocket . . . . .	215
lcc::CANGridConnectOverTcp . . . . .	220
lcc::CANGridConnectOverUSBSerial . . . . .	224
lcc::CANHeader . . . . .	229
lcc::CanMessage . . . . .	233
lcc::CanTransport . . . . .	239
FCFSupport::Car . . . . .	241
FCFSupport::CarGroup . . . . .	264
FCFSupport::CarType . . . . .	267
cmri::CMri . . . . .	276
CmriSupport::CmriNode . . . . .	283
CTCPanel::CodeButton . . . . .	291
xpressnet::CommandStationResponse . . . . .	294
xpressnet::CommandStationStatus . . . . .	296
Parsers::TrackGraph::CompressedEdgeValues . . . . .	299
Parsers::TrackGraph::CompressedNodeValues . . . . .	300
lcc::ConfigMemory . . . . .	303
lcc::ConfigOptions . . . . .	311
lcc::ConfigurationEditor . . . . .	315

Parsers::CornuBody	339
Parsers::CornuBodyElt	342
CTCPanel::Crossing	349
CTCPanel::Crossover	353
FCFSupport::PDFFileStructures::CrossReferenceTable	357
CTCPanel::CTCLabel	361
CTCPanel::CTCPanel	365
ctiacela::CTIAcela	390
CTCPanel::CurvedBlock	394
Instruments::DialInstrument	398
Instruments::DigitalClock	404
Instruments::DigitalInstrument	406
FCFSupport::Division	409
xpressnet::DoubleHeaderInformation	415
xpressnet::DoubleHeaderMuError	422
CTCPanel::DoubleSlip	423
Parsers::TrackGraph::EdgeValues	428
CTCPanel::EndBumper	430
TTSupport::eqstr	433
lcc::EventID	434
lcc::EventID_or_null	437
lcc::EventLog	438
lcc::EventReceived	441
xpressnet::FunctionStatus	447
linuxgpio::GpioInputActiveHigh	452
linuxgpio::GpioInputActiveLow	454
linuxgpio::GpioOutputSafeHigh	456
linuxgpio::GpioOutputSafeHighInvert	458
linuxgpio::GpioOutputSafeLow	461
linuxgpio::GpioOutputSafeLowInverted	463
lcc::GridConnectMessage	465
lcc::GridConnectReply	471
TTSupport::hash	478
CTCPanel::HiddenBlock	479
HTMLHelp::HTMLHelp	483
FCFSupport::PDFFileStructures::IndirectObject	515
FCFSupport::PDFFileStructures::Dictionary	402
FCFSupport::PDFFileStructures::IndirectObjectDictionary	521
FCFSupport::PDFFileStructures::InformationDirectory	538
FCFSupport::PDFFileStructures::PageLabelTree	741
FCFSupport::PDFFileStructures::ResourceDictionary	844
FCFSupport::PDFFileStructures::TypedDictionary	1247
FCFSupport::PDFFileStructures::CatalogDictionary	272
FCFSupport::PDFFileStructures::FontDictionary	443
FCFSupport::PDFFileStructures::Type1FontDictionary	1242
FCFSupport::PDFFileStructures::PostScriptStandardType1FontDictionary	799
FCFSupport::PDFFileStructures::Page	734
FCFSupport::PDFFileStructures::PageLabelDictionary	738
FCFSupport::PDFFileStructures::PageTree	745
FCFSupport::PDFFileStructures::FreedObject	446
FCFSupport::PDFFileStructures::IndirectFloatVector	513
FCFSupport::PDFFileStructures::PDFStream	786
FCFSupport::PDFFileStructures::Rectangle	840
FCFSupport::Industry	524

Parsers::IntegerList	542
CTCPanel::Lamp	546
xpressnet::LI100Message	565
xpressnet::LI100VersionNumbers	566
xpressnet::LI101XPressNetAddress	568
linuxgpio::LinuxGpio	570
xpressnet::LocomotiveAddress	573
CabWidgets::LocomotiveDirection	575
xpressnet::LocomotiveInformation	579
CabWidgets::LocomotiveSpeed	587
FCFSupport::LogMessageCallback	594
mainwindow	604
lcc::MTIDetail	635
lcc::MTIHeader	641
nce::NCE	645
lcc::nid_or_null	672
Parsers::TrackGraph::NodeValues	673
TTSupport::Occupied	678
lcc::OpenLCBMessage	684
lcc::OpenLCBNode	687
lcc::OpenLCBOverTcp	700
lcc::OpenLCBProtocols	707
ostringstream	
FCFSupport::PDFFileStructures::PDFStream	786
OvalWidgets::OvalButton	709
OvalWidgets::OvalRoundCornerRectangle	712
OvalWidgets::OvalScale	713
OvalWidgets::OvalSlider	718
OvalWidgets::OvalScrollBar	723
FCFSupport::Owner	730
Parsers::ParseFile	749
Parsers::LayoutFile	550
Parsers::MRRXtrkCad	629
ParseXML	753
FCFSupport::PathName	758
TTSupport::PathName	766
FCFSupport::PauseCallback	775
Parsers::TrackGraph::Point	788
Parsers::TurnoutBodyElt::Pos	789
Parsers::BezierBodyElt::Pos	790
Parsers::CornuBodyElt::Pos	791
FCFSupport::PrinterDevice	801
FCFSupport::LQ24PrinterDevice	597
FCFSupport::PDFPrinterDevice	777
FCFSupport::PostScriptPrinterDevice	792
FCFSupport::TextPrinterDevice	1111
CTCPanel::PushButton	810
raildriver::RaildriverClient	813
RaildriverIO	817
Parsers::RouteVec	851
Satellite	852
CTCPanel::SchLabel	854
CTCPanel::ScissorCrossover	858
ScrollTabNotebook	862

Parsers::SegPos . . . . .	878
Parsers::SegVector . . . . .	879
CabWidgets::SelectLocomotive . . . . .	883
lcc::SendEvent . . . . .	886
xpressnet::ServiceModeResponse . . . . .	888
FCFSupport::ShowBannerCallback . . . . .	890
CTCPanel::Signal . . . . .	892
CTCPanel::SIGPlate . . . . .	897
SimpleDOMEElement . . . . .	902
CTCPanel::SingleSlip . . . . .	909
xpressnet::SoftwareVersion . . . . .	921
splash . . . . .	923
azatrax::Azatrax::StateDataPacket . . . . .	937
FCFSupport::Station . . . . .	941
TTSupport::Station . . . . .	946
FCFSupport::SwitchListElement::StationOrIndustry . . . . .	953
FCFSupport::Train::StationOrIndustry . . . . .	954
TTSupport::StationTimes . . . . .	955
azatrax::SR4::status1_union . . . . .	958
azatrax::SL2::status1_union . . . . .	961
azatrax::MRD::status1_union . . . . .	963
azatrax::MRD::status2_union . . . . .	965
azatrax::SR4::status2_union . . . . .	967
azatrax::SL2::status2_union . . . . .	969
azatrax::SR4::status3_union . . . . .	971
azatrax::SL2::status3_union . . . . .	973
TTSupport::Stop . . . . .	975
TTSupport::StorageTrack . . . . .	985
CTCPanel::StraightBlock . . . . .	992
CTCPanel::StubYard . . . . .	997
CTCPanel::Switch . . . . .	1001
FCFSupport::SwitchList . . . . .	1005
FCFSupport::SwitchListElement . . . . .	1012
CTCPanel::SWPlate . . . . .	1019
FCFSupport::System . . . . .	1023
CTCPanel::ThreeWaySW . . . . .	1115
CTCPanel::ThroughYard . . . . .	1120
TTSupport::TimeRange . . . . .	1124
TTSupport::TimeTableSystem . . . . .	1130
CTCPanel::Toggle . . . . .	1154
Parsers::TrackBody . . . . .	1160
Parsers::TrackBodyElt . . . . .	1163
Parsers::TrackGraph . . . . .	1167
TTSupport::Train . . . . .	1196
FCFSupport::Train . . . . .	1206
FCFSupport::TrainDisplayCallback . . . . .	1220
Parsers::TrackGraph::Transform2D . . . . .	1223
Parsers::TurnoutBody . . . . .	1227
Parsers::TurnoutBodyElt . . . . .	1231
Parsers::TurnoutGraphic . . . . .	1239
Parsers::TurnoutRoutelist . . . . .	1241
vector	
FCFSupport::PDFFileStructures::IndirectFloatVector . . . . .	513
FCFSupport::PDFFileStructures::PDFNameArray . . . . .	776

FCFSupport::WorkInProgressCallback . . . . .	1250
xpressnet::XPressNet . . . . .	1252
xpressnet::XpressNetEvent . . . . .	1273
YY_MRRXtrkCad_INHERIT . . . . .	1275
yy_MRRXtrkCad_stype . . . . .	1280
yyItype . . . . .	1283



## Chapter 5

# Class Index

### 5.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">xpressnet::AccessoryDecoderInformation</a>	
Accessory decoder information . . . . .	161
<a href="#">Instruments::AnalogClock</a>	
Analog clock instrument . . . . .	166
<a href="#">azatrax::Azatrax</a>	
Azatrax I/O Class . . . . .	168
<a href="#">Parsers::BezierBody</a>	
List of Bezier body lines (T, E, S, and C lines) . . . . .	182
<a href="#">Parsers::BezierBodyElt</a>	
Bezier Body elements: T, E, S, and C lines are collected . . . . .	185
<a href="#">TTSupport::Cab</a>	
This class maintains information about cabs . . . . .	192
<a href="#">Instruments::CabSignalLamp</a>	
Cab signal lamp type . . . . .	196
<a href="#">lcc::CanAlias</a>	
Implements a CAN Alias . . . . .	200
<a href="#">lcc::CANGridConnect</a>	
Base class to connect to a CAN bus using GridConnect formatted message over . . . . .	203
<a href="#">lcc::CANGridConnectOverCANSocket</a>	
Connect to a CAN bus using GridConnect formatted message over a CAN Socket connection . . . . .	215
<a href="#">lcc::CANGridConnectOverTcp</a>	
Connect to a CAN bus using GridConnect formatted message over a Tcp/Ip connection . . . . .	220
<a href="#">lcc::CANGridConnectOverUSBSerial</a>	
Connect to a CAN bus using GridConnect formatted message over a USB Serial port . . . . .	224
<a href="#">lcc::CANHeader</a>	
CAN Header type . . . . .	229
<a href="#">lcc::CanMessage</a>	
A CAN Message, containing a 29-bit header and upto 8 bytes of data . . . . .	233
<a href="#">lcc::CanTransport</a>	
Logical transport of CAN Messages . . . . .	239
<a href="#">FCFSupport::Car</a>	
This class holds all of the information for a single car . . . . .	241

<a href="#">FCFSupport::CarGroup</a>	
Car group class	264
<a href="#">FCFSupport::CarType</a>	
Type of railroad car (rolling stock)	267
<a href="#">FCFSupport::PDFFileStructures::CatalogDictionary</a>	
Master catalog of the PDF file	272
<a href="#">cmri::CMri</a>	
Main C/MRI interface class	276
<a href="#">CmriSupport::CmriNode</a>	
CMR/I node type	283
<a href="#">CTCPanel::CodeButton</a>	
Code button object type	291
<a href="#">xpressnet::CommandStationResponse</a>	
General response class	294
<a href="#">xpressnet::CommandStationStatus</a>	
Command station status	296
<a href="#">Parsers::TrackGraph::CompressedEdgeValues</a>	
Compressed graph edge values	299
<a href="#">Parsers::TrackGraph::CompressedNodeValues</a>	
Compressed graph node values	300
<a href="#">lcc::ConfigMemory</a>	
Configure memory	303
<a href="#">lcc::ConfigOptions</a>	
Display memory config options	311
<a href="#">lcc::ConfigurationEditor</a>	
Generate OpenLCB Memory Configuration Window	315
<a href="#">Parsers::CornuBody</a>	
List of Cornu body lines (T, E, S, and C lines)	339
<a href="#">Parsers::CornuBodyElt</a>	
Cornu Body elements: T, E, S, and C lines are collected	342
<a href="#">CTCPanel::Crossing</a>	
Crossing object type	349
<a href="#">CTCPanel::Crossover</a>	
Crossover (turnout) object type	353
<a href="#">FCFSupport::PDFFileStructures::CrossReferenceTable</a>	
The cross reference table object	357
<a href="#">CTCPanel::CTCLabel</a>	
CTC Label object type	361
<a href="#">CTCPanel::CTCPanel</a>	
Main CTC Panel megawidget	365
<a href="#">ctiacela::CTIAcela</a>	
Main CTIAcela interface class	390
<a href="#">CTCPanel::CurvedBlock</a>	
Curved Block object type	394
<a href="#">Instruments::DialInstrument</a>	
Generic dial instrument	398
<a href="#">FCFSupport::PDFFileStructures::Dictionary</a>	
PDF Dictionary class	402
<a href="#">Instruments::DigitalClock</a>	
Digital clock instrument	404
<a href="#">Instruments::DigitalInstrument</a>	
Digital instrument	406
<a href="#">FCFSupport::Division</a>	
Implements a single division, which contains a number of contiguous stations	409



<a href="#">xpressnet::DoubleHeaderInformation</a>	
Double header information	415
<a href="#">xpressnet::DoubleHeaderMuError</a>	
Double header or MU error	422
<a href="#">CTCPanel::DoubleSlip</a>	
Double Slip (turnout) object type	423
<a href="#">Parsers::TrackGraph::EdgeValues</a>	
Uncompressed graph edge values	428
<a href="#">CTCPanel::EndBumper</a>	
End Bumper object type	430
<a href="#">TTSupport::eqstr</a>	433
<a href="#">lcc::EventID</a>	
An event id structure	434
<a href="#">lcc::EventID_or_null</a>	
An <a href="#">EventID</a> or empty string	437
<a href="#">lcc::EventLog</a>	
Event received log, with event sender	438
<a href="#">lcc::EventReceived</a>	
Display a received event	441
<a href="#">FCFSupport::PDFFileStructures::FontDictionary</a>	
A Font dictionary object	443
<a href="#">FCFSupport::PDFFileStructures::FreedObject</a>	
A deleted indirect object	446
<a href="#">xpressnet::FunctionStatus</a>	
Function status	447
<a href="#">linuxgpio::GpioInputActiveHigh</a>	
Input pin, active high (high is true)	452
<a href="#">linuxgpio::GpioInputActiveLow</a>	
Input pin, active low (low is true)	454
<a href="#">linuxgpio::GpioOutputSafeHigh</a>	
Output pin, initialized to high	456
<a href="#">linuxgpio::GpioOutputSafeHighInvert</a>	
Output pin, initialized to high, inverted	458
<a href="#">linuxgpio::GpioOutputSafeLow</a>	
Output pin, initialized to low	461
<a href="#">linuxgpio::GpioOutputSafeLowInverted</a>	
Output pin, initialized to low, with inverted logic	463
<a href="#">lcc::GridConnectMessage</a>	
A Grid Connect formatted CAN message	465
<a href="#">lcc::GridConnectReply</a>	
A Grid Connect formatted CAN message (reply)	471
<a href="#">TTSupport::hash</a>	
Option hash map, used for Print options	478
<a href="#">CTCPanel::HiddenBlock</a>	
Hidden Block object type	479
<a href="#">HTMLHelp::HTMLHelp</a>	
A widget that implements a help dialog that renders HTML coded help pages (generally generated from LaTeX using tex4ht's hlatex script)	483
<a href="#">FCFSupport::PDFFileStructures::IndirectFloatVector</a>	
Indirect array of floats	513
<a href="#">FCFSupport::PDFFileStructures::IndirectObject</a>	
Indirect object base class	515
<a href="#">FCFSupport::PDFFileStructures::IndirectObjectDictionary</a>	
PDF Indirect Object <a href="#">Dictionary</a> , used for named resources in a Resource <a href="#">Dictionary</a>	521

<a href="#">FCFSupport::Industry</a>	
<a href="#">Industry</a>	524
<a href="#">FCFSupport::PDFFileStructures::InformationDirectory</a>	
<a href="#">Information directory</a>	538
<a href="#">Parsers::IntegerList</a>	
The <a href="#">IntegerList</a> class implements a linked list of integers, used for turnout route lists	542
<a href="#">CTCPanel::Lamp</a>	
<a href="#">Lamp object type</a>	546
<a href="#">Parsers::LayoutFile</a>	
File to parse an XTrkCad layout file and create a track graph	550
<a href="#">xpressnet::LI100Message</a>	
<a href="#">LI100 messages</a>	565
<a href="#">xpressnet::LI100VersionNumbers</a>	
<a href="#">LI100 Version Numbers</a>	566
<a href="#">xpressnet::LI101XPressNetAddress</a>	
<a href="#">LI101 XPress Net Address</a>	568
<a href="#">linuxgpio::LinuxGpio</a>	
Base generic GPIO interface class	570
<a href="#">xpressnet::LocomotiveAddress</a>	
<a href="#">Locomotive address</a>	573
<a href="#">CabWidgets::LocomotiveDirection</a>	
<a href="#">Locomotive Direction widget</a>	575
<a href="#">xpressnet::LocomotiveInformation</a>	
<a href="#">Locomotive information</a>	579
<a href="#">CabWidgets::LocomotiveSpeed</a>	
<a href="#">Locomotive Speed widget</a>	587
<a href="#">FCFSupport::LogMessageCallback</a>	
A callback to log a message	594
<a href="#">FCFSupport::LQ24PrinterDevice</a>	
Class for an LQ24 compatible printer	597
<a href="#">mainwindow</a>	
A widget that is heavily extended from the BWidget MainFrame windget	604
<a href="#">azatrax::MRD</a>	
<a href="#">MRD I/O Class</a>	621
<a href="#">Parsers::MRRXtrkCad</a>	
<a href="#">MRRXtrkCad</a> parser class	629
<a href="#">lcc::MTIDetail</a>	
MTI Header type, detailed version	635
<a href="#">lcc::MTIHeader</a>	
MTI Header type	641
<a href="#">nce::NCE</a>	
Main <a href="#">NCE</a> Cab Bus interface class	645
<a href="#">lcc::nid_or_null</a>	
Node ID regexp pattern or the empty string	672
<a href="#">Parsers::TrackGraph::NodeValues</a>	
Uncompressed graph node values	673
<a href="#">TTSupport::Occupied</a>	
This class records a train sitting on a storage track during a specified time frame	678
<a href="#">lcc::OpenLCBMessage</a>	
<a href="#">OpenLCB Message type</a>	684
<a href="#">lcc::OpenLCBNode</a>	
Connect to a <a href="#">OpenLCB</a> interface	687
<a href="#">lcc::OpenLCBOverTcp</a>	
Connect to a <a href="#">OpenLCB</a> over <a href="#">Tcp</a> / <a href="#">Ip</a>	700

<a href="#">Icc::OpenLCBProtocols</a>	
Supported LCC Protocol name type	707
<a href="#">OvalWidgets::OvalButton</a>	
Oval button	709
<a href="#">OvalWidgets::OvalRoundCornerRectangle</a>	
Oval Round Corner Rectangle	712
<a href="#">OvalWidgets::OvalScale</a>	
An oval scale widget, much like a standard Tk scale widget	713
<a href="#">OvalWidgets::OvalSlider</a>	
Oval Slider	718
<a href="#">OvalWidgets::OvalScrollBar</a>	
Oval ScrollBar	723
<a href="#">FCFSupport::Owner</a>	
Describes a car owner	730
<a href="#">FCFSupport::PDFFileStructures::Page</a>	
Describes a single page	734
<a href="#">FCFSupport::PDFFileStructures::PageLabelDictionary</a>	
Page label dictionary	738
<a href="#">FCFSupport::PDFFileStructures::PageLabelTree</a>	
A tree of page label dictionaries	741
<a href="#">FCFSupport::PDFFileStructures::PageTree</a>	
A tree of pages	745
<a href="#">Parsers::ParseFile</a>	
Virtual base class for file-based parsers	749
<a href="#">ParseXML</a>	
Class to hold an XML tree	753
<a href="#">FCFSupport::PathName</a>	
A Class that portably represents a pathname	758
<a href="#">TTSupport::PathName</a>	
A Class that portably represents a pathname	766
<a href="#">FCFSupport::PauseCallback</a>	
The Pause callback	775
<a href="#">FCFSupport::PDFFileStructures::PDFNameArray</a>	
PDF Name array	776
<a href="#">FCFSupport::PDFPrinterDevice</a>	
PDF Printer device	777
<a href="#">FCFSupport::PDFFileStructures::PDFStream</a>	
Stream object	786
<a href="#">Parsers::TrackGraph::Point</a>	
Position structure	788
<a href="#">Parsers::TurnoutBodyElt::Pos</a>	
Position structure	789
<a href="#">Parsers::BezierBodyElt::Pos</a>	
Position structure	790
<a href="#">Parsers::CornuBodyElt::Pos</a>	
Position structure	791
<a href="#">FCFSupport::PostScriptPrinterDevice</a>	
Derived class for printing on Postscript printers	792
<a href="#">FCFSupport::PDFFileStructures::PostScriptStandardType1FontDictionary</a>	
A standard Type1 PostScript font dictionary	799
<a href="#">FCFSupport::PrinterDevice</a>	
Base class for printer devices (hard copy output)	801
<a href="#">CTCPanel::PushButton</a>	
Push Button object type	810

<a href="#">raildriver::RaildriverClient</a>	
Raildriver Client class – connects to the Raildriver daemon	813
<a href="#">RaildriverIO</a>	
Low-level Raildriver I/O functions	817
<a href="#">FCFSupport::PDFFileStructures::Rectangle</a>	
A rectangle object	840
<a href="#">FCFSupport::PDFFileStructures::ResourceDictionary</a>	
Resource dictionary	844
<a href="#">Parsers::RouteVec</a>	
Route structure	851
<a href="#">Satellite</a>	
Satellite class	852
<a href="#">CTCPanel::SchLabel</a>	
Schematic Label object type	854
<a href="#">CTCPanel::ScissorCrossover</a>	
Scissor <a href="#">Crossover</a> (turnout) object type	858
<a href="#">ScrollTabNotebook</a>	
Tabbed Notebook with scrollable tabs	862
<a href="#">Parsers::SegPos</a>	
Segment position, endpoint or other coordinate	878
<a href="#">Parsers::SegVector</a>	
Segemnt structure	879
<a href="#">CabWidgets::SelectLocomotive</a>	
Select or enter a Locomotive address	883
<a href="#">lcc::SendEvent</a>	
Send Event Dialog – send PCRE message	886
<a href="#">xpressnet::ServiceModeResponse</a>	
Service mode response	888
<a href="#">FCFSupport::ShowBannerCallback</a>	
Display a page heading type message on the screen	890
<a href="#">CTCPanel::Signal</a>	
Signal object type	892
<a href="#">CTCPanel::SIGPlate</a>	
Signal plate object type	897
<a href="#">SimpleDOMEElement</a>	
A simple DOM element coded in Tcl using SNIT	902
<a href="#">CTCPanel::SingleSlip</a>	
Single Slip (turnout) object type	909
<a href="#">azatrax::SL2</a>	
SL2 I/O Class	913
<a href="#">xpressnet::SoftwareVersion</a>	
Software version	921
<a href="#">splash</a>	
Widget that implements a spash window	923
<a href="#">azatrax::SR4</a>	
SR4 I/O Class	928
<a href="#">azatrax::Azatrax::StateDataPacket</a>	
Raw USB Data Packet	937
<a href="#">FCFSupport::Station</a>	
Implements a single station	941
<a href="#">TTSupport::Station</a>	
Implements a station	946

<a href="#">FCFSupport::SwitchListElement::StationOrIndustry</a>	
A const pointer to a train's stop, which can be either a station or an industry, depending on what kind of train it is	953
<a href="#">FCFSupport::Train::StationOrIndustry</a>	
Union of stations or industries, used for stops	954
<a href="#">TTSupport::StationTimes</a>	
Station times class, used by the LaTeX generator methods	955
<a href="#">azatrax::SR4::status1_union</a>	
Status byte 1 union type (Outputs)	958
<a href="#">azatrax::SL2::status1_union</a>	
Status byte 1 union type (Output states)	961
<a href="#">azatrax::MRD::status1_union</a>	
Status byte 1 union type	963
<a href="#">azatrax::MRD::status2_union</a>	
Status byte 2 union type	965
<a href="#">azatrax::SR4::status2_union</a>	
Status byte 2 union type (Input sense)	967
<a href="#">azatrax::SL2::status2_union</a>	
Status byte 2 union type (Input sense)	969
<a href="#">azatrax::SR4::status3_union</a>	
Status byte 3 union type (Input Control Status)	971
<a href="#">azatrax::SL2::status3_union</a>	
Status byte 3 union type (Input control state)	973
<a href="#">TTSupport::Stop</a>	
This class implements a stop	975
<a href="#">TTSupport::StorageTrack</a>	
Implements a storage track	985
<a href="#">CTCPanel::StraightBlock</a>	
Straight Block object type	992
<a href="#">CTCPanel::StubYard</a>	
Stub Yard object type	997
<a href="#">CTCPanel::Switch</a>	
Switch (turnout) object type	1001
<a href="#">FCFSupport::SwitchList</a>	
The global switch list structure	1005
<a href="#">FCFSupport::SwitchListElement</a>	
This class implements each switch list element	1012
<a href="#">CTCPanel::SWPlate</a>	
Switch plate object type	1019
<a href="#">FCFSupport::System</a>	
This is the main Freight Car Forwarder class	1023
<a href="#">FCFSupport::TextPrinterDevice</a>	
Derived class for printing on generic plain text printers	1111
<a href="#">CTCPanel::ThreeWaySW</a>	
Three Way Switch (turnout) object type	1115
<a href="#">CTCPanel::ThroughYard</a>	
Through Yard object type	1120
<a href="#">TTSupport::TimeRange</a>	
Implements a range of times	1124
<a href="#">TTSupport::TimeTableSystem</a>	
This is the main Time Table Class	1130
<a href="#">CTCPanel::Toggle</a>	
Toggle switch object type	1154

<a href="#">Parsers::TrackBody</a>	
List of track endpoints (T and E lines)	. 1160
<a href="#">Parsers::TrackBodyElt</a>	
Track endpoint elements (T and E lines)	. 1163
<a href="#">Parsers::TrackGraph</a>	
Track Graph class, which encapsulates the track graphs	. 1167
<a href="#">TTSupport::Train</a>	
This class implements a train	. 1196
<a href="#">FCFSupport::Train</a>	
Train	. 1206
<a href="#">FCFSupport::TrainDisplayCallback</a>	
Callback to manage a train status display	. 1220
<a href="#">Parsers::TrackGraph::Transform2D</a>	
Two dimensional transform class	. 1223
<a href="#">Parsers::TurnoutBody</a>	
List of turnout body lines (T, E, P, S, C, and J lines)	. 1227
<a href="#">Parsers::TurnoutBodyElt</a>	
Turnout body elements: T, E, P, S, C, and J lines are collected	. 1231
<a href="#">Parsers::TurnoutGraphic</a>	
Structure holding a turnout's graphical information	. 1239
<a href="#">Parsers::TurnoutRoutelist</a>	
Turnout route list structure	. 1241
<a href="#">FCFSupport::PDFFileStructures::Type1FontDictionary</a>	
Type 1 Font dictionary	. 1242
<a href="#">FCFSupport::PDFFileStructures::TypedDictionary</a>	
Typed dictionary	. 1247
<a href="#">FCFSupport::WorkInProgressCallback</a>	
Work In Progress Callback	. 1250
<a href="#">xpressnet::XPressNet</a>	
Main <a href="#">XPressNet</a> interface class	. 1252
<a href="#">xpressnet::XpressNetEvent</a>	
XPressNet Event class	. 1273
<a href="#">YY_MRRXtrkCad_INHERIT</a>	. 1275
<a href="#">yy_MRRXtrkCad_stype</a>	. 1280
<a href="#">yyItype</a>	. 1283

## Chapter 6

# File Index

### 6.1 File List

Here is a list of all files with brief descriptions:

C++/Azatrax/ <a href="#">Azatrax.h</a>	1285
C++/Azatrax/ <a href="#">mrd.h</a>	1286
C++/Azatrax/ <a href="#">sl2.h</a>	1286
C++/Azatrax/ <a href="#">sr4.h</a>	1287
C++/FCFSupport/ <a href="#">CallBack.h</a>	1287
C++/FCFSupport/ <a href="#">Car.h</a>	1288
C++/FCFSupport/ <a href="#">CarType.h</a>	1288
C++/FCFSupport/ <a href="#">Division.h</a>	1289
C++/FCFSupport/ <a href="#">FCFSupportGroup.h</a>	1290
C++/FCFSupport/ <a href="#">Industry.h</a>	1290
C++/FCFSupport/ <a href="#">LQ24Printer.h</a>	1290
C++/FCFSupport/ <a href="#">Owner.h</a>	1291
C++/FCFSupport/ <a href="#">PathName.h</a>	1291
C++/FCFSupport/ <a href="#">PDFPrinter.h</a>	1292
C++/FCFSupport/ <a href="#">PDFPrinterSupport.h</a>	1293
C++/FCFSupport/ <a href="#">PostScriptPrinter.h</a>	1295
C++/FCFSupport/ <a href="#">Printer.h</a>	1296
C++/FCFSupport/ <a href="#">Station.h</a>	1296
C++/FCFSupport/ <a href="#">SwitchList.h</a>	1297
C++/FCFSupport/ <a href="#">System.h</a>	1298
C++/FCFSupport/ <a href="#">TextPrinter.h</a>	1299
C++/FCFSupport/ <a href="#">Train.h</a>	1299
C++/ParserClasses/ <a href="#">BezierBody.h</a>	1300
C++/ParserClasses/ <a href="#">CornuBody.h</a>	1302
C++/ParserClasses/ <a href="#">IntegerList.h</a>	1303
C++/ParserClasses/ <a href="#">MRRXtrkCad.tab.h</a>	1303
C++/ParserClasses/ <a href="#">ParseFile.h</a>	1311
C++/ParserClasses/ <a href="#">ParserClassesGroup.h</a>	1311
C++/ParserClasses/ <a href="#">SocketPair.h</a>	1312
C++/ParserClasses/ <a href="#">TrackBody.h</a>	1312
C++/ParserClasses/ <a href="#">TrackGraph.h</a>	1312

C++/ParserClasses/ <a href="#">TurnoutBody.h</a>	1313
C++/RailDriver/ <a href="#">RaildriverIO.h</a>	1314
C++/TclSocketCAN/ <a href="#">TclSocketCAN.i</a>	1315
C++/TTSupport/ <a href="#">Cab.h</a>	1315
C++/TTSupport/ <a href="#">PathName.h</a>	1292
C++/TTSupport/ <a href="#">Station.h</a>	1297
C++/TTSupport/ <a href="#">TimeTableSystem.h</a>	1316
C++/TTSupport/ <a href="#">TimeTableSystemTcl.h</a>	1318
C++/TTSupport/ <a href="#">Train.h</a>	1300
C++/TTSupport/ <a href="#">TTSupportGroup.h</a>	1318
C++/wiringPi/ <a href="#">tclwiringpi.i</a>	1318
Doc/doxygen/ <a href="#">titlepage.h</a>	1320
Scripts/CMri/ <a href="#">cmri.tcl</a>	1320
Scripts/Common/ <a href="#">CabWidgets.tcl</a>	1320
Scripts/Common/ <a href="#">CommonTclGroup.h</a>	1321
Scripts/Common/ <a href="#">CTCPanel2.tcl</a>	1321
Scripts/Common/ <a href="#">fileentry.tcl</a>	1322
Scripts/Common/ <a href="#">gettext.tcl</a>	1323
Scripts/Common/ <a href="#">HTMLHelp.tcl</a>	1323
Scripts/Common/ <a href="#">labelcombobox.tcl</a>	1324
Scripts/Common/ <a href="#">labelselectcolor.tcl</a>	1325
Scripts/Common/ <a href="#">labelspinbox.tcl</a>	1325
Scripts/Common/ <a href="#">mainwindow.tcl</a>	1326
Scripts/Common/ <a href="#">panedw.tcl</a>	1326
Scripts/Common/ <a href="#">ParseXML.tcl</a>	
Parse XML and create a simple DOM tree	1326
Scripts/Common/ <a href="#">ReadConfiguration.tcl</a>	1326
Scripts/Common/ <a href="#">snitScrollNotebook.tcl</a>	1327
Scripts/Common/ <a href="#">splash.tcl</a>	1327
Scripts/ControlSupport/ <a href="#">CmriSupport.tcl</a>	1327
Scripts/CTIAcela/ <a href="#">CTIAcela.tcl</a>	1328
Scripts/GRSupport/ <a href="#">grsupport2.tcl</a>	1328
Scripts/GRSupport/ <a href="#">GRSupportTclGroup.h</a>	1329
Scripts/GRSupport/ <a href="#">Instruments2.tcl</a>	1329
Scripts/GRSupport/ <a href="#">LCARSWidgets2.tcl</a>	1330
Scripts/GRSupport/ <a href="#">OvalWidgets2.tcl</a>	1330
Scripts/LCC/ <a href="#">ConfigDialogs.tcl</a>	1331
Scripts/LCC/ <a href="#">ConfigurationEditor.tcl</a>	1332
Scripts/LCC/ <a href="#">eventDialogs.tcl</a>	1332
Scripts/LCC/ <a href="#">lcc.tcl</a>	1333
Scripts/LinuxGpio/ <a href="#">LinuxGpio.tcl</a>	1335
Scripts/NCE/ <a href="#">nce.tcl</a>	1336
Scripts/RailDriverSupport/ <a href="#">raildriver_client.tcl</a>	1337
Scripts/Satellite/ <a href="#">Satellite.tcl</a>	1338
Scripts/XPressNet/ <a href="#">xpressnet.tcl</a>	1338



## Chapter 7

# Module Documentation

### 7.1 Azatrax

Azatrax C++ LibUSB 1.0 Interface.

#### Namespaces

- [azatrax](#)  
*Azatrax C++ LibUSB 1.0 Interface.*

#### 7.1.1 Detailed Description

Azatrax C++ LibUSB 1.0 Interface.

This library implements a class that interfaces to Azatrax's MRD2-S, MRD2-U, SL2, and SR4 modules. Both the MRD2-S and the MRD2-U contain a pair a IR sensors, a stopwatch, and the MRD2-S contains relays. These units can sense trains, either reflectively (typically 'buried' in the track roadbed) or across the track. The sensors trigger the stopwatch and can also trigger the relays. The SL2 and SR4 modules are intended to operate switch machines, either directly (SL2) or via something like a NCE Switch-It (SR4). Both the SL2 and SR4 also have inputs meant for local push-button control (but they can be used for other purposes).

I was contacted by John Parsons of Azatrax (<http://www.azatrax.com>) and he kindly sent some samples of the MRD2-U and MRD2-S modules. This code is the result. The samples allowed me to test the code.

#### Author

Robert Heller <[heller@deepsoft.com](mailto:heller@deepsoft.com)>

## 7.2 FCFSupportModule

Freight Car Forwarder C++ support classes.

Freight Car Forwarder C++ support classes.

These classes implement the low-level support code for my second port of Tim O'Connor's Freight Car Forwarder system.

The main class, System, implements a complete railroad system, which consists of one or more divisions with one or more stations and industries. Running over the trackage are one or more trains, pulling an assortment of cars (some loaded and some empty). The cars are of various types, suitable for various types of loads. The system class collects the data for all of these items into one big data structure and implements the various algorithms to create a freight car forwarding system using switchlists.

The original system was written in QBASIC and was a mess of spaghetti code. I first recoded it as a pure Tcl/Tk application and because Tcl completely lacks a 'goto' statement, I needed to unravel every strand of 'spaghetti'. The Tcl code worked, but was somewhat slow. This C++ version puts the more computationally intensive (mostly heavy data indexing logic) into C++, using the STL to implement the various aggregate collections of objects. These objects are indexed and crossed indexed heavily and the forwarding algorithms traverses these collections frequently.

### Author

Robert Heller <heller@deepsoft.com>

## 7.3 ParserClasses

File-based parser classes.

### Namespaces

- [Parsers](#)

*File-based parser classes.*

### Functions

- list [tcl\\_socketpair](#) ()

*Tcl interface to socketpair.*

#### 7.3.1 Detailed Description

File-based parser classes.

These are file-based parser classes. Right now only one parser for XTrkCAD layout files. Other classes might be added later.

Included are classes used by the XTrkCAD parser. These classes are used to store the track plan information in an XTrkCAD layout file, specifically as it relates to operating issues, such as dispatching and signaling.

The track plan is loaded into a directed graph representation, where each node is one logical piece of trackwork. From this graph representation a schematic display could be created in a semi-automated way.

This package features the use of the Boost Graph Library as the underlying structure for the track graph built from reading in XTrkCAD layout files.

#### Author

Robert Heller <heller@deepsoft.com>

#### 7.3.2 Function Documentation

##### 7.3.2.1 `tcl_socketpair()`

```
list tcl_socketpair ( )
```

Tcl interface to socketpair.

Returns a list of two file channels, which are opposite ends of a connected pair of sockets.

## 7.4 TimeTableSystem

Time Table C++ support class library.

### Modules

- [TimeTableSystemTcl](#)

*Tcl Helper functions.*

### Namespaces

- [TTSupport](#)

*Time Table Support Namespace.*

### 7.4.1 Detailed Description

Time Table C++ support class library.

This class library handles all of the low-level data structures and processing for the TimeTable (V2) program. This includes the representation of scheduled trains, the stations they stop at (or go by), and the data needed to generate formatted and printed timetables.

A time table system consists of a vector of stations, which are in the order that the stations exist along the track. There is a map of cabs, a map of trains, a vector of notes, and a hash table of print options also stored in a train system. In addition, there are some system wide scalar parameter settings.

The Time Table class includes code to read and write itself to a specially formatted text file for storage between editing or processing sessions. The class includes a method to generate a LaTeX file, which can be processed by LaTeX to create a formatted timetable which can be printed. It is assumed that the `TimeTable.sty` is available for inclusion by the LaTeX system.

The ideas and structure of this code was heavily influenced by Bruce Chubb's Kalmbach book, How to Operate Your Model Railroad.

#### Author

Robert Heller <heller@deepsoft.com>

## 7.5 TimeTableSystemTcl

Tcl Helper functions.

### Functions

- TimeTableSystem \* [NewCreateTimeTable](#) (const char \*name, int timescale, int timeinterval)  
*Tcl constructor to create a new TimeTable.*
- TimeTableSystem \* [OldCreateTimeTable](#) (const char \*filename, char \*\*outmessage)  
*Tcl constructor to create a time table system from an existing file.*
- int [ForEveryStation](#) (Tcl\_Interp \*interp, TimeTableSystem \*timetable, Tcl\_Obj \*variable, Tcl\_Obj \*body)  
*Tcl looping construct for Stations.*
- int [ForEveryCab](#) (Tcl\_Interp \*interp, TimeTableSystem \*timetable, Tcl\_Obj \*variable, Tcl\_Obj \*body)  
*Tcl looping construct for Cabs.*
- int [ForEveryTrain](#) (Tcl\_Interp \*interp, TimeTableSystem \*timetable, Tcl\_Obj \*variable, Tcl\_Obj \*body)  
*Tcl looping construct for Trains.*
- int [ForEveryNote](#) (Tcl\_Interp \*interp, TimeTableSystem \*timetable, Tcl\_Obj \*variable, Tcl\_Obj \*body)  
*Tcl looping construct for notes.*
- int [ForEveryPrintOption](#) (Tcl\_Interp \*interp, TimeTableSystem \*timetable, Tcl\_Obj \*variable, Tcl\_Obj \*body)  
*Tcl looping construct for print options.*
- int [TT\\_StringListToList](#) (Tcl\_Interp \*interp, const char \*stringList)  
*Tcl function to convert a serialized string list to a Tcl list.*
- int [TT\\_ListToStringListString](#) (Tcl\_Interp \*interp, Tcl\_Obj \*list)  
*Tcl function to convert a Tcl list to a serialized string list.*

### Variables

- apply int [Tcl\\_Result](#) { int [TTSupport::ForEveryStation](#) }

#### 7.5.1 Detailed Description

Tcl Helper functions.

These are top level Tcl support functions for the TimeTableSystem class. They are only available from Tcl, C++ programs have other API functions, including overloaded constructors and iterator methods.

#### Author

Robert Heller <heller@deepsoft.com>

#### 7.5.2 Function Documentation

### 7.5.2.1 ForEveryCab()

```
int ForEveryCab (
    Tcl_Interp * interp,
    TimeTableSystem * timetable,
    Tcl_Obj * variable,
    Tcl_Obj * body )
```

Tcl looping construct for Cabs.

Tcl looping construct that loops over the cabs in timetable, setting variable to the Cab pointer and evaluates body.

#### Parameters

<i>timetable</i>	The time table object.
<i>variable</i>	The loop variable.
<i>body</i>	The body script.

#### Returns

An empty string.

### 7.5.2.2 ForEveryNote()

```
int ForEveryNote (
    Tcl_Interp * interp,
    TimeTableSystem * timetable,
    Tcl_Obj * variable,
    Tcl_Obj * body )
```

Tcl looping construct for notes.

Tcl looping construct that loops over the notes in timetable, setting variable to the note string and evaluates body.

#### Parameters

<i>timetable</i>	The time table object.
<i>variable</i>	The loop variable.
<i>body</i>	The body script.

#### Returns

An empty string.

### 7.5.2.3 ForEveryPrintOption()

```
int ForEveryPrintOption (
    Tcl_Interp * interp,
    TimeTableSystem * timetable,
    Tcl_Obj * variable,
    Tcl_Obj * body )
```

Tcl looping construct for print options.

Tcl looping construct that loops over the stations in timetable, setting variable to the print option key and evaluates body.

#### Parameters

<i>timetable</i>	The time table object.
<i>variable</i>	The loop variable.
<i>body</i>	The body script.

#### Returns

An empty string.

### 7.5.2.4 ForEveryStation()

```
int ForEveryStation (
    Tcl_Interp * interp,
    TimeTableSystem * timetable,
    Tcl_Obj * variable,
    Tcl_Obj * body )
```

Tcl looping construct for Stations.

Tcl looping construct that loops over the stations in timetable, setting variable to the Station pointer and evaluates body.

#### Parameters

<i>timetable</i>	The time table object.
<i>variable</i>	The loop variable.
<i>body</i>	The body script.

#### Returns

An empty string.

### 7.5.2.5 ForEveryTrain()

```
int ForEveryTrain (
    Tcl_Interp * interp,
    TimeTableSystem * timetable,
    Tcl_Obj * variable,
    Tcl_Obj * body )
```

Tcl looping construct for Trains.

Tcl looping construct that loops over the stations in timetable, setting variable to the Train pointer and evaluates body.

#### Parameters

<i>timetable</i>	The time table object.
<i>variable</i>	The loop variable.
<i>body</i>	The body script.

#### Returns

An empty string.

### 7.5.2.6 NewCreateTimeTable()

```
TimeTableSystem* NewCreateTimeTable (
    const char * name,
    int timescale,
    int timeinterval )
```

Tcl constructor to create a new TimeTable.

Tcl constructor to create a new TimeTable. Calls the new time table constructor.

#### Parameters

<i>name</i>	The name of the time table system.
<i>timescale</i>	Number of time units per 24 hours. There are 1440 minutes in 24 hours.
<i>timeinterval</i>	The tick frequency in time units.

#### Returns

A TimeTableSystem object.



### 7.5.2.7 OldCreateTimeTable()

```
TimeTableSystem* OldCreateTimeTable (
    const char * filename,
    char ** outmessage )
```

Tcl constructor to create a time table system from an existing file.

Tcl constructor to create a time table system from an existing file. The file is read in and the class is properly initialized from the data in the file.

#### Parameters

<i>filename</i>	The name of the file to load.
-----------------	-------------------------------

#### Returns

A TimeTableSystem object.

### 7.5.2.8 TT\_ListToStringListString()

```
int TT_ListToStringListString (
    Tcl_Interp * interp,
    Tcl_Obj * list )
```

Tcl function to convert a Tcl list to a serialized string list.

Used to convert Tcl lists to a form that the C++ code can deal with portably.

#### Parameters

<i>list</i>	A Tcl list.
-------------	-------------

#### Returns

A serialized string list.

### 7.5.2.9 TT\_StringListToList()

```
int TT_StringListToList (
    Tcl_Interp * interp,
    const char * stringList )
```

Tcl function to convert a serialized string list to a Tcl list.

Used to convert serialized C++ string lists to a Tcl list.

#### Parameters

<i>stringList</i>	A serialized string list.
-------------------	---------------------------

#### Returns

A Tcl list.

### 7.5.3 Variable Documentation

#### 7.5.3.1 Tcl\_Result

```
apply int Tcl_Result { int TTSupport::ForEveryStation }
```

## 7.6 Station

[Station](#) and support classes.

### Classes

- class [TTSupport::Occupied](#)  
*This class records a train sitting on a storage track during a specified time frame.*
- class [TTSupport::TimeRange](#)  
*The [TimeRange](#) class implements a range of times.*
- class [TTSupport::StorageTrack](#)  
*The [StorageTrack](#) class implements a storage track.*
- class [TTSupport::Station](#)  
*The [Station](#) class implements a station.*

### Typedefs

- typedef map< [TimeRange](#), [Occupied](#), less< [TimeRange](#) > > [TTSupport::OccupiedMap](#)  
*The [Occupied](#) Map type, ordered by time ranges.*
- typedef map< string, [StorageTrack](#), less< string > > [TTSupport::StorageTrackMap](#)  
*Storage track map.*
- typedef vector< [Station](#) > [TTSupport::StationVector](#)  
*[Station](#) Vector.*

#### 7.6.1 Detailed Description

[Station](#) and support classes.

This class and its support classes implement information about stations and station stops. This includes where a station is along the line (its scale mile), what storage tracks it has, and what trains are being stored on the storage tracks during what times. Stations are places where trains stop or just important junctions or mile post locations that trains pass by – they might only be used for time keeping checks. Note: the classification tracks at a yard are not storage tracks. Nor are RIP or service or other special purpose tracks. Storage tracks are only for storing whole, complete trains (they might be without engines).

#### Author

Robert Heller <heller@deepsoft.com>

#### 7.6.2 Typedef Documentation

### 7.6.2.1 OccupiedMap

```
typedef map<TimeRange, Occupied, less<TimeRange> > TTSupport::OccupiedMap
```

The `Occupied` Map type, ordered by time ranges.

#### Author

Robert Heller <heller@deepsoft.com>

### 7.6.2.2 StationVector

```
typedef vector<Station> TTSupport::StationVector
```

`Station` Vector.

### 7.6.2.3 StorageTrackMap

```
typedef map<string, StorageTrack, less<string> > TTSupport::StorageTrackMap
```

Storage track map.

Indexed by name.

#### Author

Robert Heller <heller@deepsoft.com>

## 7.7 Train and support classes.

The train class implements a running train and lists the stations it passes (and possibly stops at).

### Classes

- class `TTSupport::Stop`  
*This class implements a stop.*
- class `TTSupport::Train`  
*This class implements a train.*

### Typedefs

- typedef vector< `Stop` > `TTSupport::StopVector`  
*A vector of stops.*
- typedef map< string, `Train` \*, less< string > > `TTSupport::TrainNumberMap`  
*`Train` number map, indexed by train number (symbol).*

#### 7.7.1 Detailed Description

The train class implements a running train and lists the stations it passes (and possibly stops at).

#### Author

Robert Heller <heller@deepsoft.com>

#### 7.7.2 Typedef Documentation

##### 7.7.2.1 StopVector

```
typedef vector<Stop> TTSupport::StopVector
```

A vector of stops.

##### 7.7.2.2 TrainNumberMap

```
typedef map<string, Train *, less<string> > TTSupport::TrainNumberMap
```

`Train` number map, indexed by train number (symbol).

## 7.8 Cab

[Cab](#) class and support types.

### Classes

- class [TTSupport::Cab](#)  
*This class maintains information about cabs.*

### Typedefs

- typedef map< string, [Cab](#) \*, less< string > > [TTSupport::CabNameMap](#)  
*[Cab](#) name map, cabs indexed by name.*

#### 7.8.1 Detailed Description

[Cab](#) class and support types.

This only really important for pure DC systems, but it useful for DCC systems as a way define crew(s).

#### Author

Robert Heller <heller@deepsoft.com>

#### 7.8.2 Typedef Documentation

##### 7.8.2.1 CabNameMap

```
typedef map<string, Cab *, less<string> > TTSupport::CabNameMap
```

[Cab](#) name map, cabs indexed by name.

## 7.9 TclSocketCANModule

Tcl Channel for CAN Sockets.

### Functions

- SWIGEXPORT int [Tclsocketcan\\_SafeInit](#) (Tcl\_Interp \*)
- int [SocketCAN](#) (Tcl\_Interp \*interp, const char \*candev)

*Open a CAN Socket.*

### Variables

- module [TclSocketCAN](#)
- include typemaps [i](#)

### 7.9.1 Detailed Description

Tcl Channel for CAN Sockets.

This module defines a Tcl Channel type for CAN Sockets. These are much like TCP Sockets, except the read/write code translates CAN frames to/from GridConnect messages. This module is only available under Linux, since only Linux has kernel support for the CAN socket family (AF\_CAN).

#### Author

Robert Heller <heller@deepsoft.com>

### 7.9.2 Function Documentation

#### 7.9.2.1 SocketCAN()

```
int SocketCAN (  
    Tcl_Interp * interp,  
    const char * candev )
```

Open a CAN Socket.

This function opens a read/write connection to a CAN socket to the named interface. The result of this function is the name of a Tcl Channel and can be used as an argument to any Tcl Channel function (such as gets, puts, or fileevent).



## Parameters

<i>candev</i>	The name of the CAN interface to connect to.
---------------	--

## Returns

The name of a Tcl Channel.

## 7.9.2.2 Tclsocketcan\_SafeInit()

```
SWIGEXPORT int Tclsocketcan_SafeInit (
    Tcl_Interp * )
```

## 7.9.3 Variable Documentation

## 7.9.3.1 i

```
include typemaps i
```

## Initial value:

```
{
#define SWIG_name
#define SWIG_version
%}

%apply int Tcl_Result { int SocketCAN }
```

Referenced by TTSupport::Stop::AddNote(), TTSupport::Train::AddNoteToTrain(), TTSupport::TimeTableSystem::AddStorageTrack(), FCFSupport::System::CarTypesOrder(), TTSupport::TimeTableSystem::DuplicateStationIndex(), TTSupport::TimeTableSystem::FindStorageTrack(), FCFSupport::Industry::IncrementStatsLen(), TTSupport::TimeTableSystem::lthStation(), TTSupport::Stop::Note(), TTSupport::Train::Note(), FCFSupport::Owner::Owner(), TTSupport::TimeTableSystem::SetDuplicateStationIndex(), TTSupport::TimeTableSystem::SMile(), TTSupport::TimeTableSystem::StationName(), TTSupport::Train::Stopl(), FCFSupport::Industry::TheCar(), FCFSupport::System::TheCar(), FCFSupport::System::TheCarGroup(), FCFSupport::System::TheDivision(), FCFSupport::Station::TheIndustry(), FCFSupport::System::TheIndustry(), FCFSupport::System::TheStation(), FCFSupport::System::TrainByIndex(), and FCFSupport::PDFFileStructures::IndirectFloatVector::WriteDirect().

## 7.9.3.2 TclSocketCAN

```
module TclSocketCAN
```

## Initial value:

```
{
    static const char rcsid[] = "@(#) : $Id$"
```

## 7.10 TcLwiringpiModule

WiringPi TcL bindings module.

WiringPi TcL bindings module.

This module provides TcL bindings to the Wiring Pi library. All of the functions defined in the Wiring Pi C library are made available as TcL commands. Please see the documentation for the Wiring Pi C library.

### Author

Robert Heller <heller@deepsoft.com>

## 7.11 Cmri

CMR/I Tcl Serial Port Interface.

### Namespaces

- [cmri](#)

*CMR/I Tcl Serial Port Interface.*

### 7.11.1 Detailed Description

CMR/I Tcl Serial Port Interface.

This is a cross-platform implementation of Bruce Chubb's C/MRI QBASIC serial port code ported to Tcl. This code has been tested with Tcl 8.4.

Basically, the way this code works is to use a SNIT class (described on in CMri) to interface to the serial port, which may have one or more serial port cards (a mix of USICs, SUSICs, and SMINIs). A given class instance interfaces to all of the cards on attached to a given serial port. There are three public member functions, one to initialize a given board (described in CMri::InitBoard), one to set the output ports (described in CMri::Outputs), and one to poll the state of the input ports (described in CMri::Inputs).

I was inspired to write this code after reading the four part series in Model Railroader and reading the download package for the SMINI card. I already have a copy of Bruce Chubb's Build Your Own Universal Computer Interface, but the SMINI looks like a great option for small "remote" locations of a layout where there are a few turnouts and a some signals, such as a small junction, interchange yard, or isolated industrial spur.

### Author

Robert Heller <heller@deepsoft.com>

## 7.12 CTIAcela

CTI Acela Serial Port Interface.

### Namespaces

- [ctiacela](#)  
*CTI Acela Tcl Serial Port Interface.*

### Classes

- class [ctiacela::CTIAcela](#)  
*Main [CTIAcela](#) interface class.*

### Typedefs

- typedef int [ctiacela::addresstype](#)  
*Module address type.*
- typedef int [ctiacela::ubyte](#)  
*Unsigned byte type.*
- typedef int [ctiacela::speedtype](#)  
*Speed type.*
- typedef int [ctiacela::momtype](#)  
*Momentum control type.*
- typedef int [ctiacela::filterthreshtype](#)  
*Filter threshold type.*

### Functions

- static [ctiacela::CTIAcela::validate](#) (object)  
*Type validation method.*
- [ctiacela::CTIAcela::CTIAcela](#) (name, port,...)  
*Constructor: open a connection to the CTI Acela.*
- [ctiacela::CTIAcela::~~CTIAcela](#) ()  
*The destructor restores the serial port's state and closes it.*
- [ctiacela::CTIAcela::HaveData](#) ()
- [ctiacela::CTIAcela::OnlineP](#) ()
- [ctiacela::CTIAcela::\\_handleSRQ](#) ()  
*Handle a service request.*
- static [ctiacela::CTIAcela::highbyte](#) (addr)  
*Return the high byte of address.*
- static [ctiacela::CTIAcela::lowbyte](#) (addr)  
*Return the low byte of address.*
- [ctiacela::CTIAcela::Activate](#) (address)

- Activate a control.*

  - `ctiacela::CTIAcela::Deactive` (address)
- Deactive a control.*

  - `ctiacela::CTIAcela::PulseOn` (address, pulsewidth)
- Pulse On a control.*

  - `ctiacela::CTIAcela::PulseOff` (address, pulsewidth)
- Pulse Off a control.*

  - `ctiacela::CTIAcela::Blink` (address, pulsewidth)
- Blink a control.*

  - `ctiacela::CTIAcela::ReverseBlink` (address, pulsewidth)
- Reverse Blink a control.*

  - `ctiacela::CTIAcela::Control4` (address, c1, c2, c3, c4)
- Configure 4 controls simultaneously.*

  - `ctiacela::CTIAcela::Control8` (address, c1, c2, c3, c4, c5, c6, c7, c8)
- Configure 8 controls simultaneously.*

  - `ctiacela::CTIAcela::Control16` (address, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12, c13, c14, c15, c16)
- Configure 16 controls simultaneously.*

  - static `ctiacela::CTIAcela::pack4` (b1, b2, b3, b4)
- pack 4 bits*

  - static `ctiacela::CTIAcela::pack8` (b1, b2, b3, b4, b5, b6, b7, b8)
- pack 4 bits*

  - `ctiacela::CTIAcela::Throttle` (address, speed, momentum, brake, direction, idle)
- Throttle command.*

  - `ctiacela::CTIAcela::EmergencyStop` ()
- Emergency Stop.*

  - `ctiacela::CTIAcela::Signal2` (address, lamp1, lamp2, yellow="off")
- Control 2-lamp signals.*

  - `ctiacela::CTIAcela::Signal3` (address, lamp1, lamp2, lamp3)
- Control 3-lamp signals.*

  - `ctiacela::CTIAcela::Signal4` (address, lamp1, lamp2, lamp3, lamp4)
- Control 4-lamp signals.*

  - `ctiacela::CTIAcela::SignalSettings` (blinkrate, yellowhue)
- Set Signal Settings.*

  - `ctiacela::CTIAcela::SignalBrightness` (brightness)
- Set signal brightness.*

  - `ctiacela::CTIAcela::ConfigureSensor` (address, threshold, select, polarity)
- Configure a sensor.*

  - `ctiacela::CTIAcela::Read` (address)
- Read the state of a sensor.*

  - `ctiacela::CTIAcela::Read4` (address)
- Read the state of four sensors.*

  - `ctiacela::CTIAcela::Read8` (address)
- Read the state of eight sensors.*

  - `ctiacela::CTIAcela::Read16` (address)
- Read the state of sixteen sensors.*

  - `ctiacela::CTIAcela::ReadAll` ()
- Read all sensors.*

- `ctiacela::CTIAcela::SRQControl` (enable="yes")  
*Enable or disable SRQ messages.*
- `ctiacela::CTIAcela::Query` ()  
*Query sensor change state.*
- `ctiacela::CTIAcela::ResetNetwork` ()  
*Reset the network.*
- `ctiacela::CTIAcela::NetworkOnline` ()  
*Bring the network online.*
- `ctiacela::CTIAcela::NetworkOffline` ()  
*Bring the network offline.*
- `ctiacela::CTIAcela::Poll` ()  
*Poll the network configuration.*
- `ctiacela::CTIAcela::ReadRevision` ()  
*Read CTI Acela firmware revision.*
- `ctiacela::CTIAcela::_transmit` (buffer, responsebytes=0)  
*Transmit buffer and wait for response.*
- `ctiacela::CTIAcela::_readevent` ()  
*Read event method.*
- `ctiacela::CTIAcela::_readbyte` (thebytevar)  
*Read a single byte from the serial interface.*

## Variables

- `ctiacela::ctiacela`
- static `ctiacela::CTIAcela::Responses`  
*Responses.*
- static `ctiacela::CTIAcela::Opcodes`  
*Opcodes.*
- static `ctiacela::CTIAcela::LampBits`  
*Lamp Bits.*
- static `ctiacela::CTIAcela::FilterSelectBits`  
*Filter Select Bits.*
- static `ctiacela::CTIAcela::CTI_DeviceMap`  
*CTI Module Map.*
- `ctiacela::CTIAcela::ttyfd`  
*Terminal file descriptor.*
- `ctiacela::CTIAcela::dataavailable`  
*Flag set to true (yes) when sensor data is available.*
- `ctiacela::CTIAcela::networkonline`  
*Flag set to false (no) when the network goes offline.*
- `ctiacela::CTIAcela::_timeout`  
*Timeout flag.*
- static `ctiacela::CTIAcela::maxtries`  
*Loop control for read attempts.*

### 7.12.1 Detailed Description

CTI Acela Serial Port Interface.

This is a cross-platform implementation of the host (computer) side of the CTI Acela Serial Port Interface.

Basically, the way this code works is to use a SNIT class (described in the CTIAcela type) to interface to the serial port, which may have one or more CTI modules (Train Brain, Dash-8, Watchman, Signalman, Smart Cab, etc.). A given class instance interfaces to all of the modules attached to a given serial port.

#### Author

Robert Heller <heller@deepsoft.com>

### 7.12.2 Typedef Documentation

#### 7.12.2.1 addresstype

```
unsigned short int ctiacela::addresstype
```

Module address type.

An integer in the range from 0 to 65535, inclusive.

#### 7.12.2.2 filterthreshtype

```
int ctiacela::filterthreshtype
```

Filter threshold type.

An integer from 0 to 31, inclusive.

#### 7.12.2.3 momtype

```
int ctiacela::momtype
```

Momentum control type.

Integer in the range of 0 to 7, inclusive.

#### 7.12.2.4 speedtype

```
int ctiacela::speedtype
```

Speed type.

Integer in the range of 0 to 100, inclusive.

#### 7.12.2.5 ubyte

```
unsigned char ctiacela::ubyte
```

Unsigned byte type.

An integer in the range from 0 to 255, inclusive.

### 7.12.3 Function Documentation

#### 7.12.3.1 \_handleSRQ()

```
ctiacela::CTIAcela::_handleSRQ ( ) [private]
```

Handle a service request.

#### 7.12.3.2 \_readbyte()

```
ctiacela::CTIAcela::_readbyte (
    thebytevar ) [private]
```

Read a single byte from the serial interface.

Used by methods that read responses.

##### Parameters

<i>thebytevar</i>	A name of a variable to put the byte read. Undefined if there was an error.
-------------------	---

##### Returns

false on error and true on success.



### 7.12.3.3 \_readevent()

```
ctiacela::CTIAcela::_readevent ( ) [private]
```

Read event method.

### 7.12.3.4 \_transmit()

```
ctiacela::CTIAcela::_transmit (
    buffer ,
    responsebytes = 0 ) [private]
```

Transmit buffer and wait for response.

#### Parameters

<i>buffer</i>	List of bytes to transmit.
<i>responsebytes</i>	Number of expected databytes (not counting the command ack byte) or N for a variable number of result bytes.

#### Returns

the response, either a single byte or a list of bytes.

### 7.12.3.5 Activate()

```
ctiacela::CTIAcela::Activate (
    address )
```

Activate a control.

#### Parameters

<i>address</i>	Address of the control.
----------------	-------------------------

### 7.12.3.6 Blink()

```
ctiacela::CTIAcela::Blink (
    address ,
    pulsewidth )
```

Blink a control.

#### Parameters

<i>address</i>	Address of the control.
<i>pulsewidth</i>	Pulsewidth in 10ths of a second

### 7.12.3.7 ConfigureSensor()

```
ctiacela::CTIAcela::ConfigureSensor (
    address ,
    threshold ,
    select ,
    polarity )
```

Configure a sensor.

#### Parameters

<i>address</i>	Address of the sensor.
<i>threshold</i>	Filter threshold, 0-31.
<i>select</i>	Filter select, one of noise, bounce, gap, pr dirty.
<i>polarity</i>	Polarity, one of normal or invert.

### 7.12.3.8 Control16()

```
ctiacela::CTIAcela::Control16 (
    address ,
    c1 ,
    c2 ,
    c3 ,
    c4 ,
    c5 ,
    c6 ,
    c7 ,
    c8 ,
```

```

c9 ,
c10 ,
c11 ,
c12 ,
c13 ,
c14 ,
c15 ,
c16 )

```

Configure 16 controls simultaneously.

#### Parameters

<i>address</i>	Address of the first control.
<i>c1</i>	First control status, boolean true activates, boolean false deactivates.
<i>c2</i>	Second control status, boolean true activates, boolean false deactivates.
<i>c3</i>	Third control status, boolean true activates, boolean false deactivates.
<i>c4</i>	Fourth control status, boolean true activates, boolean false deactivates.
<i>c5</i>	Fifth control status, boolean true activates, boolean false deactivates.
<i>c6</i>	Sixth control status, boolean true activates, boolean false deactivates.
<i>c7</i>	Seventh control status, boolean true activates, boolean false deactivates.
<i>c8</i>	Eighth control status, boolean true activates, boolean false deactivates.
<i>c9</i>	Ninth control status, boolean true activates, boolean false deactivates.
<i>c10</i>	Tenth control status, boolean true activates, boolean false deactivates.
<i>c11</i>	Eleventh control status, boolean true activates, boolean false deactivates.
<i>c12</i>	Twelfth control status, boolean true activates, boolean false deactivates.
<i>c13</i>	Thirteenth control status, boolean true activates, boolean false deactivates.
<i>c14</i>	Fourteenth control status, boolean true activates, boolean false deactivates.
<i>c15</i>	Fifteenth control status, boolean true activates, boolean false deactivates.
<i>c16</i>	Sixteenth control status, boolean true activates, boolean false deactivates.

#### 7.12.3.9 Control4()

```

ctiacela::CTIAcela::Control4 (
    address ,
    c1 ,
    c2 ,
    c3 ,
    c4 )

```

Configure 4 controls simultaneously.

#### Parameters

<i>address</i>	Address of the first control.
<i>c1</i>	First control status, boolean true activates, boolean false deactivates.
<i>c2</i>	Second control status, boolean true activates, boolean false deactivates.
<i>c3</i>	Third control status, boolean true activates, boolean false deactivates.
<i>c4</i>	Fourth control status, boolean true activates, boolean false deactivates.

### 7.12.3.10 Control8()

```
ctiacela::CTIAcela::Control8 (
    address ,
    c1 ,
    c2 ,
    c3 ,
    c4 ,
    c5 ,
    c6 ,
    c7 ,
    c8 )
```

Configure 8 controls simultaneously.

#### Parameters

<i>address</i>	Address of the first control.
<i>c1</i>	First control status, boolean true activates, boolean false deactivates.
<i>c2</i>	Second control status, boolean true activates, boolean false deactivates.
<i>c3</i>	Third control status, boolean true activates, boolean false deactivates.
<i>c4</i>	Fourth control status, boolean true activates, boolean false deactivates.
<i>c5</i>	Fifth control status, boolean true activates, boolean false deactivates.
<i>c6</i>	Sixth control status, boolean true activates, boolean false deactivates.
<i>c7</i>	Seventh control status, boolean true activates, boolean false deactivates.
<i>c8</i>	Eighth control status, boolean true activates, boolean false deactivates.

### 7.12.3.11 CTIAcela()

```
ctiacela::CTIAcela::CTIAcela (
    name ,
    port ,
    ... )
```

Constructor: open a connection to the CTI Acela.

#### Parameters

<i>name</i>	The name of this instance.
<i>port</i>	Name of the serial port connected to theCTI Acela.
...	Options: <ul style="list-style-type: none"> <li>• -srqhandler Script to run when there is a sense state change.</li> </ul>

#### 7.12.3.12 Deactive()

```
ctiacela::CTIAcela::Deactive (
    address )
```

Deactive a control.

##### Parameters

<i>address</i>	Address of the control.
----------------	-------------------------

#### 7.12.3.13 EmergencyStop()

```
ctiacela::CTIAcela::EmergencyStop ( )
```

Emergency Stop.

Stop all trains.

#### 7.12.3.14 HaveData()

```
ctiacela::CTIAcela::HaveData ( )
```

##### Returns

Yes, if there is data available.

#### 7.12.3.15 highbyte()

```
static ctiacela::CTIAcela::highbyte (
    addr ) [static], [private]
```

Return the high byte of address.

**Parameters**

<i>addr</i>	Address word (16-bits)
-------------	------------------------

**Returns**

upper 8 bits

**7.12.3.16 lowbyte()**

```
static ctiacela::CTIAcela::lowbyte (  
    addr ) [static], [private]
```

Return the low byte of address.

**Parameters**

<i>addr</i>	Address word (16-bits)
-------------	------------------------

**Returns**

lower 8 bits

**7.12.3.17 NetworkOffline()**

```
ctiacela::CTIAcela::NetworkOffline ( )
```

Bring the network offline.

**7.12.3.18 NetworkOnline()**

```
ctiacela::CTIAcela::NetworkOnline ( )
```

Bring the network online.

### 7.12.3.19 OnlineP()

```
ctiacela::CTIAcela::OnlineP ( )
```

#### Returns

Yes, if the network is online.

### 7.12.3.20 pack4()

```
static ctiacela::CTIAcela::pack4 (
    b1 ,
    b2 ,
    b3 ,
    b4 ) [static], [private]
```

pack 4 bits

#### Parameters

<i>b1</i>	First bit
<i>b2</i>	Second bit
<i>b3</i>	Third bit
<i>b4</i>	Fourth bit

#### Returns

a byte with the bits packed.

### 7.12.3.21 pack8()

```
static ctiacela::CTIAcela::pack8 (
    b1 ,
    b2 ,
    b3 ,
    b4 ,
    b5 ,
    b6 ,
    b7 ,
    b8 ) [static], [private]
```

pack 4 bits

**Parameters**

<i>b1</i>	First bit
<i>b2</i>	Second bit
<i>b3</i>	Third bit
<i>b4</i>	Fourth bit
<i>b5</i>	Fifth bit
<i>b6</i>	Sixth bit
<i>b7</i>	Seventh bit
<i>b8</i>	Eighth bit

**Returns**

a byte with the bits packed.

**7.12.3.22 Poll()**

```
ctiacela::CTIAcela::Poll ( )
```

Poll the network configuration.

**Returns**

a list of modules on the network.

**7.12.3.23 PulseOff()**

```
ctiacela::CTIAcela::PulseOff (
    address ,
    pulsewidth )
```

Pulse Off a control.

**Parameters**

<i>address</i>	Address of the control.
<i>pulsewidth</i>	Pulsewidth in 10ths of a second



#### 7.12.3.24 PulseOn()

```
ctiacela::CTIAcela::PulseOn (
    address ,
    pulsewidth )
```

Pulse On a control.

##### Parameters

<i>address</i>	Address of the control.
<i>pulsewidth</i>	Pulsewidth in 10ths of a second

#### 7.12.3.25 Query()

```
ctiacela::CTIAcela::Query ( )
```

Query sensor change state.

##### Returns

true if sensors changed state since the last Query call.

#### 7.12.3.26 Read()

```
ctiacela::CTIAcela::Read (
    address )
```

Read the state of a sensor.

##### Parameters

<i>address</i>	Address of the sensor.
----------------	------------------------

##### Returns

the sensor state as a boolean value.

#### 7.12.3.27 Read16()

```
ctiacela::CTIAcela::Read16 (  
    address )
```

Read the state of sixteen sensors.

##### Parameters

<i>address</i>	Address of the first sensor.
----------------	------------------------------

##### Returns

the state of sixteen sensors as a sixteen element list of boolean values.

#### 7.12.3.28 Read4()

```
ctiacela::CTIAcela::Read4 (  
    address )
```

Read the state of four sensors.

##### Parameters

<i>address</i>	Address of the first sensor.
----------------	------------------------------

##### Returns

the state of four sensors as a four element list of boolean values.

#### 7.12.3.29 Read8()

```
ctiacela::CTIAcela::Read8 (  
    address )
```

Read the state of eight sensors.

##### Parameters

<i>address</i>	Address of the first sensor.
----------------	------------------------------

**Returns**

the state of eight sensors as an eight element list of boolean values.

**7.12.3.30 ReadAll()**

```
ctiacela::CTIAcela::ReadAll ( )
```

Read all sensors.

**Returns**

the state of all sensors as a list of boolean values.

**7.12.3.31 ReadRevision()**

```
ctiacela::CTIAcela::ReadRevision ( )
```

Read CTI Acela firmware revision.

**Returns**

a two element list containing the major and minor revision numbers of the Acela

**7.12.3.32 ResetNetwork()**

```
ctiacela::CTIAcela::ResetNetwork ( )
```

Reset the network.

**7.12.3.33 ReverseBlink()**

```
ctiacela::CTIAcela::ReverseBlink (
    address ,
    pulsewidth )
```

Reverse Blink a control.

## Parameters

<i>address</i>	Address of the control.
<i>pulsewidth</i>	Pulsewidth in 10ths of a second

## 7.12.3.34 Signal2()

```
ctiacela::CTIAcela::Signal2 (
    address ,
    lamp1 ,
    lamp2 ,
    yellow = "off" )
```

Control 2-lamp signals.

## Parameters

<i>address</i>	Address of first lamp.
<i>lamp1</i>	Lamp 1 control, one of off, on, blink, or reverseblink.
<i>lamp2</i>	Lamp 2 control, one of off, on, blink, or reverseblink.
<i>yellow</i>	Yellow control, one of off, on, blink, or reverseblink.

## 7.12.3.35 Signal3()

```
ctiacela::CTIAcela::Signal3 (
    address ,
    lamp1 ,
    lamp2 ,
    lamp3 )
```

Control 3-lamp signals.

## Parameters

<i>address</i>	Address of first lamp.
<i>lamp1</i>	Lamp 1 control, one of off, on, blink, or reverseblink.
<i>lamp2</i>	Lamp 2 control, one of off, on, blink, or reverseblink.
<i>lamp3</i>	Lamp 3control, one of off, on, blink, or reverseblink.

## 7.12.3.36 Signal4()

```
ctiacela::CTIAcela::Signal4 (
    address ,
    lamp1 ,
    lamp2 ,
    lamp3 ,
    lamp4 )
```

Control 4-lamp signals.

## Parameters

<i>address</i>	Address of first lamp.
<i>lamp1</i>	Lamp 1 control, one of off, on, blink, or reverseblink.
<i>lamp2</i>	Lamp 2 control, one of off, on, blink, or reverseblink.
<i>lamp3</i>	Lamp 3 control, one of off, on, blink, or reverseblink.
<i>lamp4</i>	Lamp 4 control, one of off, on, blink, or reverseblink.

## 7.12.3.37 SignalBrightness()

```
ctiacela::CTIAcela::SignalBrightness (
    brightness )
```

Set signal brightness.

## Parameters

<i>brightness</i>	Signal brightness.
-------------------	--------------------

## 7.12.3.38 SignalSettings()

```
ctiacela::CTIAcela::SignalSettings (
    blinkrate ,
    yellowhue )
```

Set Signal Settings.

## Parameters

<i>blinkrate</i>	Blink rate in 10ths of a second
<i>yellowhue</i>	Mix of red and green to get yellow as a percentage of green vs red: 128 is 50/50.

### 7.12.3.39 SRQControl()

```
ctiacela::CTIAcela::SRQControl (
    enable = "yes" )
```

Enable or disable SRQ messages.

#### Parameters

<i>enable</i>	Boolean, if true enable SRQ messages.
---------------	---------------------------------------

### 7.12.3.40 Throttle()

```
ctiacela::CTIAcela::Throttle (
    address ,
    speed ,
    momentum ,
    brake ,
    direction ,
    idle )
```

Throttle command.

#### Parameters

<i>address</i>	Address of the throttle.
<i>speed</i>	Speed (0-100).
<i>momentum</i>	Momentum Control (0 minimum, 7 maximum).
<i>brake</i>	Brake control (boolean: true is on).
<i>direction</i>	Direction control (forward or reverse).
<i>idle</i>	Idle Voltage Control (boolean: true is on).

### 7.12.3.41 validate()

```
static ctiacela::CTIAcela::validate (
    object ) [static]
```

Type validation method.

Validate object as a [CTIAcela](#) instance.

## Parameters

<i>object</i>	The object to validate.
---------------	-------------------------

## 7.12.3.42 ~CTIAcela()

```
ctiacela::CTIAcela::~~CTIAcela ( )
```

The destructor restores the serial port's state and closes it.

## 7.12.4 Variable Documentation

## 7.12.4.1 \_timeout

```
ctiacela::CTIAcela::_timeout [private]
```

Timeout flag.

## 7.12.4.2 CTI\_DeviceMap

```
ctiacela::CTIAcela::CTI_DeviceMap [static], [private]
```

CTI Module Map.

## 7.12.4.3 ctiacela

```
ctiacela::ctiacela
```

## 7.12.4.4 dataavailable

```
ctiacela::CTIAcela::dataavailable [private]
```

Flag set to true (yes) when sensor data is available.

#### 7.12.4.5 FilterSelectBits

```
ctiacela::CTIAcela::FilterSelectBits [static], [private]
```

Filter Select Bits.

#### 7.12.4.6 LampBits

```
ctiacela::CTIAcela::LampBits [static], [private]
```

Lamp Bits.

#### 7.12.4.7 maxtries

```
ctiacela::CTIAcela::maxtries [static], [private]
```

Loop control for read attempts.

#### 7.12.4.8 networkonline

```
ctiacela::CTIAcela::networkonline [private]
```

Flag set to false (no) when the network goes offline.

#### 7.12.4.9 Opcodes

```
ctiacela::CTIAcela::Opcodes [static], [private]
```

Opcodes.

#### 7.12.4.10 Responses

```
ctiacela::CTIAcela::Responses [static], [private]
```

Responses.

#### 7.12.4.11 ttyfd

```
ctiacela::CTIAcela::ttyfd [private]
```

Terminal file descriptor.



## 7.13 LCCModule

LCC (OpenLCB) interface code.

### Namespaces

- [lcc](#)

*Namespace that holds the LCC interface code.*

### 7.13.1 Detailed Description

LCC (OpenLCB) interface code.

These are Tcl SNIT classes that interface to the LCC / OpenLCB bus.

#### Author

Robert Heller <heller@deepsoft.com>

## 7.14 XPressNetModule

XPressNet interface code.

### Namespaces

- [xpressnet](#)

*Namespace that holds the [XPressNet](#) interface code.*

### 7.14.1 Detailed Description

XPressNet interface code.

These are Tcl SNIT classes that interface to the Lenz XPressNet interface used on Lenz DCC Command Units. There is a low-level collection of Tcl SNIT classes that handles the low-level Serial I/O interface and there is a higher level interface that defines a Tcl Event to handle the asynchronous aspects of the low-level XPressNet serial I/O, by entering the Lenz XPressnet interface into Tcl's Event processing system.

#### Author

Robert Heller <heller@deepsoft.com>

## 7.15 NCEModule

NCE Cab Bus interface code.

### Namespaces

- [nce](#)

*Namespace that holds the [NCE](#) interface code.*

### 7.15.1 Detailed Description

NCE Cab Bus interface code.

This is the Tcl SNIT class that interfaces with the NCE Cab Bus. It works with either the NCE USB Interface board (typically with the Power Cab) OR the NCE RS232 interface (typically used with the CS02 command station).

#### Author

Robert Heller <[heller@deepsoft.com](mailto:heller@deepsoft.com)>

## 7.16 TclCommon

Common Tcl Script Library.

### Namespaces

- [FileEntry](#)  
*This is a specialized form of the [LabelEntry](#) widget intended for selecting file names.*
- [ReadConfiguration](#)  
*The Read Configuration File code is contained in this namespace.*
- [CTCPanel](#)  
*CTC Panel code, Version 2.*
- [LabelSelectColor](#)  
*This package provides a BWidget style megawidget for selecting colors, in the same style as a [LabelEntry](#) widget.*
- [LabelComboBox](#)  
*This is a specialized form of the [LabelFrame](#) widget containing a [ComboBox](#) Widget.*
- [LabelSpinBox](#)  
*This is a specialized form of the [LabelFrame](#) widget containing a [SpinBox](#) Widget.*
- [PanedWindow](#)  
*A modified version of the BWidget [PanedWindow](#).*
- [CabWidgets](#)  
*Cab Widget code.*
- [HTMLHelp](#)  
*[HTMLHelp](#) namespace, which contains the [HTMLHelp](#) snit widget adapter object and associated code.*
- [gettext](#)  
*Localization functions.*

### Classes

- class [splash](#)  
*Widget that implements a splash window.*
- class [mainwindow](#)  
*A widget that is heavily extended from the BWidget [MainFrame](#) widget.*
- class [SimpleDOMElement](#)  
*A simple DOM element coded in Tcl using SNIT.*
- class [ParseXML](#)  
*Class to hold an XML tree.*
- class [ScrollTabNotebook](#)  
*Tabbed Notebook with scrollable tabs.*

### 7.16.1 Detailed Description

Common Tcl Script Library.

This script library contains a collection of Tcl Scripts that are commonly used throughout the Model Railroad System. Most of them relate to various extended GUI elements.

#### Author

Robert Heller <heller@deepsoft.com>

## 7.17 LinuxGpio

Linux GPIO interface, using sysfs.

### Namespaces

- [linuxgpio](#)

*Linux GPIO Interface.*

### 7.17.1 Detailed Description

Linux GPIO interface, using sysfs.

This is the portable implementation of GPIO under Linux, using the sysfs file system (/sys/class/gpio/...). This code should work on all SBC / development boards that run Linux (Raspberry Pis, Beagle Bones, Banana Pis, etc.).

#### Author

Robert Heller <heller@deepsoft.com>

## 7.18 Cmri Support code

### Namespaces

- [CmriSupport](#)

*Cmri Support code.*

### 7.18.1 Detailed Description

## 7.19 RaildriverClientModule

Raildriver Client class code.

### Namespaces

- [raildriver](#)

*Namespace that holds the Raildriver Client class code.*

### 7.19.1 Detailed Description

Raildriver Client class code.

This is the Tcl SNIT class that implements a client that connects to the RailDriver daemon.

#### Author

Robert Heller <heller@deepsoft.com>

## 7.20 Graphics Support Code

These scripts provide support for additional graphical objects, including (dashboard) instruments and touch screen widgets (similar to what is used on Star Trek).

These scripts provide support for additional graphical objects, including (dashboard) instruments and touch screen widgets (similar to what is used on Star Trek).

At present little of this code is actually used by any of the Model Railroad System programs, but it is provided for future use and for use by people developing and exploring additional graphical user interface options.



## 7.21 GRSupportModule

Graphics Support code.

### Namespaces

- [GRSupport](#)  
*Code to support the various graphics packages.*
- [Instruments](#)  
*Namespace used for instruments code.*
- [LCARS](#)  
*Namespace where the [LCARS](#) code lives.*
- [OvalWidgets](#)  
*These oval shaped widgets are much like the Star Trek NG computer screens.*

### 7.21.1 Detailed Description

Graphics Support code.

A collection of packages to help with various graphics needs.

#### Author

Robert Heller <heller@deepsoft.com>

## 7.22 SatelliteModule

[Satellite](#) master client.

### Classes

- class [Satellite](#)  
*[Satellite](#) class.*

### 7.22.1 Detailed Description

[Satellite](#) master client.

This Tcl SNIT class interfaces with SatelliteDaemon processes (daemons) running on slave computers (usually Raspberry Pis).

#### Author

Robert Heller <heller@deepsoft.com>

## 7.23 FCFSupport

Various callback classes.

### Namespaces

- [FCFSupport](#)

*Namespace to hold all of the FCF Support code.*

### 7.23.1 Detailed Description

Various callback classes.

Switch List Support code.

These classes are used to provide a means for various class members to access code in the outer application to handle message passing and related activities. For the most part, the base classes don't do anything at all, but provide a set of virtual methods that implement the various sorts of callback functionality.

#### Author

Robert Heller <heller@deepsoft.com>

These classes provide support to create switch lists for trains and yards.



## Chapter 8

# Namespace Documentation

### 8.1 azatrax Namespace Reference

[Azatrax](#) C++ LibUSB 1.0 Interface.

#### Classes

- class [Azatrax](#)  
*[Azatrax](#) I/O Class.*
- class [MRD](#)  
*[MRD](#) I/O Class.*
- class [SL2](#)  
*[SL2](#) I/O Class.*
- class [SR4](#)  
*[SR4](#) I/O Class.*

#### 8.1.1 Detailed Description

[Azatrax](#) C++ LibUSB 1.0 Interface.

This is the basic cross-platform class library that uses the libusb-1.0 API to communicate with MRD2-U, MRD2-S, [SL2](#), and [SR4](#) modules from [Azatrax](#) over the USB bus. This library contains classes that encapsulate logic to communicate with these devices. Each class instance connects to a specific device instance.

#### Author

Robert Heller <[heller@deepsoft.com](mailto:heller@deepsoft.com)>

#### 8.1.2 Tcl Package Provided

[Azatrax](#) 1.0.0

### 8.1.3 Library Provided

libAzatrax 1.0.0

## 8.2 CabWidgets Namespace Reference

Cab Widget code.

### Classes

- class [LocomotiveDirection](#)  
*Locomotive Direction widget.*
- class [LocomotiveSpeed](#)  
*Locomotive Speed widget.*
- class [SelectLocomotive](#)  
*Select or enter a Locomotive address.*

### 8.2.1 Detailed Description

Cab Widget code.

Contains various widgets related to DCC (or CD) Cab panels, including locomotive speed, direction, locomotive address, and DCC Programming features.

#### Author

Robert Heller <heller@deepsoft.com>

### 8.2.2 Package provided

[CabWidgets](#) 1.0

## 8.3 cmri Namespace Reference

CMR/I Tcl Serial Port Interface.

### Classes

- class [CMri](#)  
*Main C/MRI interface class.*

## Typedefs

- typedef int [uatype](#)  
*Board address type.*
- typedef int [ubyte](#)  
*Unsigned byte.*
- typedef listtype [ByteList](#)  
*List of bytes.*

## Enumerations

- enum [CardType](#) { [USIC](#), [SUSIC](#), [SMINI](#) }  
*Card type codes.*

### 8.3.1 Detailed Description

CMR/I Tcl Serial Port Interface.

This is a cross-platform implementation of Bruce Chubb's C/MRI QBASIC serial port code ported to Tcl. This code has been tested with Tcl 8.4.

Basically, the way this code works is to use a SNIT class (described on in [CMri](#)) to interface to the serial port, which may have one or more serial port cards (a mix of USICs, SUSICs, and SMINIs). A given class instance interfaces to all of the cards on attached to a given serial port. There are three public member functions, one to initialize a given board (described in [CMri::InitBoard](#)), one to set the output ports (described in [CMri::Outputs](#)), and one to poll the state of the input ports (described in [CMri::Inputs](#)).

I was inspired to write this code after reading the four part series in Model Railroader and reading the download package for the SMINI card. I already have a copy of Bruce Chubb's Build Your Own Universal Computer Interface, but the SMINI looks like a great option for small "remote" locations of a layout where there are a few turnouts and a some signals, such as a small junction, interchange yard, or isolated industrial spur.

#### Author

Robert Heller <[heller@deepsoft.com](mailto:heller@deepsoft.com)>

### 8.3.2 Package provided

Cmri 2.0.0

### 8.3.3 Typedef Documentation

#### 8.3.3.1 ByteList

```
list< cmri::ubyte > cmri::ByteList
```

List of bytes.

Contains a list of unsigned bytes.

#### 8.3.3.2 uatype

```
int cmri::uatype
```

Board address type.

An integer in the range from 0 to 127, inclusive.

#### 8.3.3.3 ubyte

```
unsigned char cmri::ubyte
```

Unsigned byte.

8-bit unsigned byte.

### 8.3.4 Enumeration Type Documentation

#### 8.3.4.1 CardType

```
enum cmri::CardType
```

Card type codes.

Enumerator

USIC	Classic Universal Serial Interface Card.
SUSIC	Super Classic Universal Serial Interface Card.
SMINI	SMINI Super Mini node.



## 8.4 CmriSupport Namespace Reference

Cmri Support code.

### Classes

- class [CmriNode](#)  
*CMR/I node type.*

### 8.4.1 Detailed Description

Cmri Support code.

This is high-level code to support the CMR/I code, in the form of a SNIT type object that wraps the low-level class and creates a network of boards on the bus that the low-level class accesses.

#### Author

Robert Heller <heller@deepsoft.com>

### 8.4.2 Package provided

[CmriSupport](#) 1.2

## 8.5 CTCPanel Namespace Reference

CTC Panel code, Version 2.

### Classes

- class [CodeButton](#)  
*Code button object type.*
- class [Crossing](#)  
*Crossing object type.*
- class [Crossover](#)  
*Crossover (turnout) object type.*
- class [CTCLabel](#)  
*CTC Label object type.*
- class [CTCPanel](#)  
*Main CTC Panel megawidget.*
- class [CurvedBlock](#)  
*Curved Block object type.*

- class [DoubleSlip](#)  
*Double Slip (turnout) object type.*
- class [EndBumper](#)  
*End Bumper object type.*
- class [HiddenBlock](#)  
*Hidden Block object type.*
- class [Lamp](#)  
*Lamp object type.*
- class [PushButton](#)  
*Push Button object type.*
- class [SchLabel](#)  
*Schematic Label object type.*
- class [ScissorCrossover](#)  
*Scissor [Crossover](#) (turnout) object type.*
- class [Signal](#)  
*Signal object type.*
- class [SIGPlate](#)  
*Signal plate object type.*
- class [SingleSlip](#)  
*Single Slip (turnout) object type.*
- class [StraightBlock](#)  
*Straight Block object type.*
- class [StubYard](#)  
*Stub Yard object type.*
- class [Switch](#)  
*Switch (turnout) object type.*
- class [SWPlate](#)  
*Switch plate object type.*
- class [ThreeWaySW](#)  
*Three Way Switch (turnout) object type.*
- class [ThroughYard](#)  
*Through Yard object type.*
- class [Toggle](#)  
*Toggle switch object type.*

## Functions

- [leverMethods](#) (hasCenter)  
*Macro to add lever methods to object types.*
- [verifyDoubleMethod](#) ()  
*Macro to add a verify double method to a snit type.*
- [verifyBoolMethod](#) ()  
*Macro to add a verify boolean method to a snit type.*
- [verifyColorMethod](#) ()  
*Macro to add a verify color method to a snit type.*
- [verifyOrientation8Method](#) ()

*Macro to add a verify 8-way orientation method to a snit type.*

- [verifyPositionMethod](#) ()

*Macro to add a verify position method to a snit type.*

- [standardMethods](#) ()

*Macro to add a standard set of methods to an object type.*

- [trackworkmethods](#) ()

*Macro to include trackwork drawing methods.*

### 8.5.1 Detailed Description

CTC Panel code, Version 2.

This version of the CTC Panel code uses tile and snit to implement CTC panels and the gadgets that populate CTC panels. The CTC Panel code is contained in this namespace.

#### Author

Robert Heller <heller@deepsoft.com>

### 8.5.2 Package provided

[CTCPanel](#) 2.0

### 8.5.3 Function Documentation

#### 8.5.3.1 leverMethods()

```
CTCPanel::leverMethods (
    hasCenter )
```

Macro to add lever methods to object types.

#### Parameters

<i>hasCenter</i>	Flag to indicate if there is a center position for this object's lever.
------------------	---

#### 8.5.3.2 standardMethods()

```
CTCPanel::standardMethods ( )
```

Macro to add a standard set of methods to an object type.

#### 8.5.3.3 trackworkmethods()

```
CTCPanel::trackworkmethods ( )
```

Macro to include trackwork drawing methods.

#### 8.5.3.4 verifyBoolMethod()

```
CTCPanel::verifyBoolMethod ( )
```

Macro to add a verify boolean method to a snit type.

#### 8.5.3.5 verifyColorMethod()

```
CTCPanel::verifyColorMethod ( )
```

Macro to add a verify color method to a snit type.

#### 8.5.3.6 verifyDoubleMethod()

```
CTCPanel::verifyDoubleMethod ( )
```

Macro to add a verify double method to a snit type.

#### 8.5.3.7 verifyOrientation8Method()

```
CTCPanel::verifyOrientation8Method ( )
```

Macro to add a verify 8-way orientation method to a snit type.

#### 8.5.3.8 verifyPositionMethod()

```
CTCPanel::verifyPositionMethod ( )
```

Macro to add a verify position method to a snit type.

## 8.6 ctiacula Namespace Reference

CTI Acela Tcl Serial Port Interface.

### Classes

- class [CTIAcela](#)  
*Main [CTIAcela](#) interface class.*

### Typedefs

- typedef int [addresstype](#)  
*Module address type.*
- typedef int [ubyte](#)  
*Unsigned byte type.*
- typedef int [speedtype](#)  
*Speed type.*
- typedef int [momtype](#)  
*Momentum control type.*
- typedef int [filterthreshtype](#)  
*Filter threshold type.*

### Variables

- [ctiacela](#)

#### 8.6.1 Detailed Description

CTI Acela Tcl Serial Port Interface.

#### Author

Robert Heller <heller@deepsoft.com>

### 8.6.2 Package provided

CTIAcela 1.0.0

## 8.7 FCFSupport Namespace Reference

Namespace to hold all of the FCF Support code.

### Namespaces

- [PDFFileStructures](#)  
*PDF File support structures.*

### Classes

- class [Car](#)  
*This class holds all of the information for a single car.*
- class [CarGroup](#)  
*Car group class.*
- class [CarType](#)  
*The CarType class represents a type of railroad car (rolling stock).*
- class [Division](#)  
*The Division class implements a single division, which contains a number of contiguous stations.*
- class [Industry](#)  
*The Industry class represents an industry.*
- class [LogMessageCallback](#)  
*A callback to log a message.*
- class [LQ24PrinterDevice](#)  
*Class for an LQ24 compatible printer.*
- class [Owner](#)  
*The Owner class describes a car owner.*
- class [PathName](#)  
*A Class that portably represents a pathname.*
- class [PauseCallback](#)  
*The Pause callback.*
- class [PDFPrinterDevice](#)  
*PDF Printer device.*
- class [PostScriptPrinterDevice](#)  
*Derived class for printing on Postscript printers.*
- class [PrinterDevice](#)  
*Base class for printer devices (hard copy output).*
- class [ShowBannerCallback](#)  
*Display a page heading type message on the screen.*
- class [Station](#)

- The [Station](#) class implements a single station.

  - class [SwitchList](#)

The global switch list structure.
  - class [SwitchListElement](#)

This class implements each switch list element.
  - class [System](#)

This is the main Freight [Car](#) Forwarder class.
  - class [TextPrinterDevice](#)

Derived class for printing on generic plain text printers.
  - class [Train](#)

The [Train](#) class represents a train.
  - class [TrainDisplayCallback](#)

Callback to manage a train status display.
  - class [WorkInProgressCallback](#)

Work In Progress Callback.

## Typedefs

- typedef vector< [Division](#) \* > [DivisionVector](#)

A vector of divisions.
- typedef map< int, [Division](#) \*, less< int > > [DivisionMap](#)

A map of divisions, by integer index (division index).
- typedef map< char, [Division](#) \*, less< char > > [DivisionSymbolMap](#)

A map of divisions, by division symbol (a character).
- typedef vector< [Station](#) \* > [StationVector](#)

A station vector.
- typedef map< int, [Station](#) \*, less< int > > [StationMap](#)

A station map by integer index.
- typedef map< int, [Train](#) \*, less< int > > [TrainMap](#)

A map of trains, indexed by integer (train index).
- typedef map< string, [Train](#) \*, less< string > > [TrainNameMap](#)

A map of trains, indexed by string (Train name).
- typedef vector< [Car](#) \* > [CarVector](#)

A vector of cars.
- typedef map< int, [Industry](#) \*, less< int > > [IndustryMap](#)

A map of industry pointers indexed by an integer.
- typedef vector< [Industry](#) \* > [IndustryVector](#)

A vector of industry pointers.
- typedef vector< char > [CarTypeOrderVector](#)

A vector of ordered car types.
- typedef map< char, [CarType](#) \*, less< char > > [CarTypeMap](#)

A map of car types indexed by type character.
- typedef map< string, [Owner](#) \*, less< string > > [OwnerMap](#)

Map of owners, indexed by their initials.
- typedef vector< [SwitchListElement](#) > [SwitchListElementVector](#)

A vector of switch list elements.
- typedef vector< string > [stringVector](#)

A Vector of strings.

## Functions

- ostream & [operator<<](#) (ostream &stream, const [SwitchListElement](#) &element)  
*Output stream operator for SwitchListElements.*
- ostream & [operator<<](#) (ostream &stream, const [PDFFileStructures::PDFNameArray](#) &parray)  
*Output stream operator for PDFNameArrays.*

### 8.7.1 Detailed Description

Namespace to hold all of the FCF Support code.

#### Author

Robert Heller <heller@deepsoft.com>

### 8.7.2 Tcl Package Provided

Fcfclasses 1.0.4

### 8.7.3 Library Provided

libfcfclasses 1.0.4

### 8.7.4 Typedef Documentation

#### 8.7.4.1 CarTypeMap

```
typedef map<char, CarType *, less<char> > FCFSupport::CarTypeMap
```

A map of car types indexed by type character.

#### 8.7.4.2 CarTypeOrderVector

```
typedef vector<char> FCFSupport::CarTypeOrderVector
```

A vector of ordered car types.



#### 8.7.4.3 CarVector

```
typedef vector< Car * > FCFSupport::CarVector
```

A vector of cars.

#### 8.7.4.4 DivisionMap

```
typedef map<int, Division *, less<int> > FCFSupport::DivisionMap
```

A map of divisions, by integer index (division index).

#### 8.7.4.5 DivisionSymbolMap

```
typedef map<char, Division *, less<char> > FCFSupport::DivisionSymbolMap
```

A map of divisions, by division symbol (a character).

#### 8.7.4.6 DivisionVector

```
typedef vector<Division *> FCFSupport::DivisionVector
```

A vector of divisions.

#### 8.7.4.7 IndustryMap

```
typedef map<int, Industry *, less<int> > FCFSupport::IndustryMap
```

A map of industry pointers indexed by an integer.

#### 8.7.4.8 IndustryVector

```
typedef vector<Industry *> FCFSupport::IndustryVector
```

A vector of industry pointers.

#### 8.7.4.9 OwnerMap

```
typedef map<string, Owner *, less<string> > FCFSupport::OwnerMap
```

Map of owners, indexed by their initials.

#### 8.7.4.10 StationMap

```
typedef map<int, Station *, less<int> > FCFSupport::StationMap
```

A station map by integer index.

#### 8.7.4.11 StationVector

```
typedef vector<Station *> FCFSupport::StationVector
```

A station vector.

#### 8.7.4.12 stringVector

```
typedef vector<string> FCFSupport::stringVector
```

A Vector of strings.

Used as the list of path list in a [PathName](#) instance.

#### Author

Robert Heller <heller@deepsoft.com>

#### 8.7.4.13 SwitchListElementVector

```
typedef vector<SwitchListElement> FCFSupport::SwitchListElementVector
```

A vector of switch list elements.

#### 8.7.4.14 TrainMap

```
typedef map<int, Train *, less<int> > FCFSupport::TrainMap
```

A map of trains, indexed by integer (train index).

#### 8.7.4.15 TrainNameMap

```
typedef map<string, Train *, less<string> > FCFSupport::TrainNameMap
```

A map of trains, indexed by string (Train name).

### 8.7.5 Function Documentation

#### 8.7.5.1 operator<<() [1/2]

```
ostream& FCFSupport::operator<< (
    ostream & stream,
    const SwitchListElement & element )
```

Output stream operator for SwitchListElements.

##### Parameters

<i>stream</i>	The output stream.
<i>element</i>	The switch list element to output.

Referenced by FCFSupport::SwitchList::ResetLastIndex().

#### 8.7.5.2 operator<<() [2/2]

```
ostream& FCFSupport::operator<< (
    ostream & stream,
    const PDFFileStructures::PDFNameArray & parray )
```

Output stream operator for PDFNameArrays.

## Parameters

<i>stream</i>	The stream to write to.
<i>parray</i>	The array to write.

## 8.8 FCFSupport::PDFFileStructures Namespace Reference

PDF File support structures.

### Classes

- class [CatalogDictionary](#)  
*Master catalog of the PDF file.*
- class [CrossReferenceTable](#)  
*The cross reference table object.*
- class [Dictionary](#)  
*PDF Dictionary class.*
- class [FontDictionary](#)  
*A Font dictionary object.*
- class [FreedObject](#)  
*A deleted indirect object.*
- class [IndirectFloatVector](#)  
*Indirect array of floats.*
- class [IndirectObject](#)  
*Indirect object base class.*
- class [IndirectObjectDictionary](#)  
*PDF Indirect Object Dictionary, used for named resources in a Resource Dictionary.*
- class [InformationDirectory](#)  
*Information directory.*
- class [Page](#)  
*Describes a single page.*
- class [PageLabelDictionary](#)  
*Page label dictionary.*
- class [PageLabelTree](#)  
*A tree of page label dictionaries.*
- class [PageTree](#)  
*A tree of pages.*
- class [PDFNameArray](#)  
*PDF Name array.*
- class [PDFStream](#)  
*Stream object.*
- class [PostScriptStandardType1FontDictionary](#)  
*A standard Type1 PostScript font dictionary.*
- class [Rectangle](#)

*A rectangle object.*

- class [ResourceDictionary](#)

*Resource dictionary.*

- class [Type1FontDictionary](#)

*Type 1 Font dictionary.*

- class [TypedDictionary](#)

*Typed dictionary.*

## Typedefs

- typedef map< string, [IndirectObject](#) \*, less< string > > [NamedIndirectObjectMap](#)

*A "vector" of named indirect objects, implemented as a map.*

- typedef vector< [PDFStream](#) \* > [PDFStreamVector](#)

*A vector of PDF Streams.*

- typedef vector< [PageLabelTree](#) \* > [PageLabelTreeKidVector](#)

*Map of [PageLabelTree](#) kids.*

- typedef map< int, [PageLabelDictionary](#) \*, less< int > > [PageLabelDictionaryNumMap](#)

*Map of [PageLabelDictionary](#) numbers.*

## Functions

- string [QuotePDFString](#) (const string &str)

*Quote a string (protect special character with a backslash).*

### 8.8.1 Detailed Description

PDF File support structures.

These classes and structures are designed to implement some of the functionality described in Adobe's PDF Reference Fifth Edition version 1.6.

#### Author

Robert Heller <heller@deepsoft.com>

### 8.8.2 Typedef Documentation

### 8.8.2.1 NamedIndirectObjectMap

```
typedef map<string, IndirectObject *, less<string> > FCFSupport::PDFFileStructures::NamedIndirectObjectMap
```

A “vector” of named indirect objects, implemented as a map.

The elements are indexed by name.

#### Author

Robert Heller <heller@deepsoft.com>

### 8.8.2.2 PageLabelDictionaryNumMap

```
typedef map<int, PageLabelDictionary*, less<int> > FCFSupport::PDFFileStructures::PageLabelDictionaryNumMap
```

Map of [PageLabelDictionary](#) numbers.

### 8.8.2.3 PageLabelTreeKidVector

```
typedef vector<PageLabelTree*> FCFSupport::PDFFileStructures::PageLabelTreeKidVector
```

Map of [PageLabelTree](#) kids.

### 8.8.2.4 PDFStreamVector

```
typedef vector<PDFStream *> FCFSupport::PDFFileStructures::PDFStreamVector
```

A vector of PDF Streams.

#### Author

Robert Heller <heller@deepsoft.com>

## 8.8.3 Function Documentation

### 8.8.3.1 QuotePDFString()

```
string FCFSupport::PDFFileStructures::QuotePDFString (
    const string & str )
```

Quote a string (protect special character with a backslash).

## Parameters

<i>str</i>	The string to quote.
------------	----------------------

## 8.9 FileEntry Namespace Reference

This is a specialized form of the LabelEntry widget intended for selecting file names.

## Functions

- [create](#) (path,...)  
*Creation procedure.*
- [configure](#) (path,...)  
*Configuration procedure: configure one or more options for this widget.*
- [cget](#) (path, option)  
*Configuration option accessor procedure: access one option directly.*
- [bind](#) (path,...)  
*Bind function.*
- [\\_path\\_command](#) (path, cmd, larg)  
*Path command for this megawidget.*
- [\\_destroy](#) (path)  
*Destructor function.*
- [\\_openFile](#) (path)  
*Prodedure bound to the file open button.*

### 8.9.1 Detailed Description

This is a specialized form of the LabelEntry widget intended for selecting file names.

A button is included to the right of the entry that pops up a file selection dialog. Many of the resources from LabelFrame, Entry, and Button are included in this widget.

## Parameters

<i>path</i>	The widget path.
-------------	------------------

## Parameters

...	<p>Options:</p> <ul style="list-style-type: none"> <li>• <code>-filebitmap</code> The name of a bitmap to use for the button. By default an option folder image is used.</li> <li>• <code>-fileimage</code> The name of an image to use for the button. By default an option folder image is used.</li> <li>• <code>-filedialog</code> The type of file dialog to use. Should be one of open, save, or directory. If open, <code>tk_getOpenFile</code> is used; if save, <code>tk_getSaveFile</code> is used; and if directory, <code>tk_chooseDirectory</code> is used.</li> <li>• <code>-defaultextension</code> This option is passed to <code>tk_getOpenFile</code> or <code>tk_getSaveFile</code>.</li> <li>• <code>-filetypes</code> This option is passed to <code>tk_getOpenFile</code> or <code>tk_getSaveFile</code>.</li> <li>• <code>-title</code> This option is passed to <code>tk_getOpenFile</code> or <code>tk_getSaveFile</code>.</li> <li>• <code>-labeljustify</code> From <code>LabelFrame</code> (<code>-justify</code>).</li> <li>• <code>-labelwidth</code> From <code>LabelFrame</code> (<code>-width</code>).</li> <li>• <code>-labelanchor</code> From <code>LabelFrame</code> (<code>-anchor</code>).</li> <li>• <code>-labelheight</code> From <code>LabelFrame</code> (<code>-height</code>).</li> <li>• <code>-labelfont</code> From <code>LabelFrame</code> (<code>-font</code>).</li> <li>• <code>-labeltextvariable</code> From <code>LabelFrame</code> (<code>-textvariable</code>).</li> <li>• <code>-label</code> From <code>LabelFrame</code> (<code>-text</code>).</li> <li>• <code>-entryfg</code> From <code>Entry</code> (<code>-foreground</code>).</li> <li>• <code>-entrybg</code> From <code>Entry</code> (<code>-background</code>).</li> <li>• <code>-text</code> From <code>Entry</code>.</li> <li>• <code>-buttonfg</code> From <code>Button</code> (<code>-foreground</code>).</li> <li>• <code>-buttonbg</code> From <code>Button</code> (<code>-background</code>).</li> <li>• <code>-buttonactivebg</code> From <code>Button</code> (<code>-activebackground</code>).</li> <li>• <code>-buttonactivefg</code> From <code>Button</code> (<code>-activeforeground</code>).</li> <li>• <code>-buttondisabledfg</code> From <code>Button</code> (<code>-disabledforeground</code>).</li> <li>• <code>-buttonhighlightbg</code> From <code>Button</code> (<code>-highlightbackground</code>).</li> <li>• <code>-buttonhighlightcolor</code> From <code>Button</code> (<code>-highlightcolor</code>).</li> </ul>
-----	--

## Author

Robert Heller <heller@deepsoft.com>

## 8.9.2 Package provided

BWFileEntry 1.0.0



### 8.9.3 Function Documentation

#### 8.9.3.1 `_destroy()`

```
FileEntry::_destroy (
    path ) [private]
```

Destructor function.

##### Parameters

<i>path</i>	The path of the megawidget.
-------------	-----------------------------

#### 8.9.3.2 `_openFile()`

```
FileEntry::_openFile (
    path ) [private]
```

Prodedure bound to the file open button.

Pops up a file selector dialog.

##### Parameters

<i>path</i>	The path of the megawidget.
-------------	-----------------------------

#### 8.9.3.3 `_path_command()`

```
FileEntry::_path_command (
    path ,
    cmd ,
    larg ) [private]
```

Path command for this megawidget.

Implements all of the megawidget commands.

**Parameters**

<i>path</i>	The path of the megawidget.
<i>cmd</i>	The command name.
<i>larg</i>	The command argument.

**8.9.3.4 bind()**

```
FileEntry::bind (
    path ,
    ... )
```

Bind function.

Passthrough to the entry widget.

**Parameters**

<i>path</i>	The path of the megawidget.
...	Bind arguments

**8.9.3.5 cget()**

```
FileEntry::cget (
    path ,
    option )
```

Configuration option accessor procedure: access one option directly.

**Parameters**

<i>path</i>	The path of the megawidget.
<i>option</i>	The option to access

**8.9.3.6 configure()**

```
FileEntry::configure (
    path ,
    ... )
```

Configuration procedure: configure one or more options for this widget.

#### Parameters

<i>path</i>	The path of the megawidget.
...	Option value pairs.

#### 8.9.3.7 create()

```
FileEntry::create (
    path ,
    ... )
```

Creation procedure.

#### Parameters

<i>path</i>	The megawidget's path.
...	Options for this widget.

## 8.10 gettext Namespace Reference

Localization functions.

### Functions

- [\\_m](#) (msgid,...)  
*Handle messages with a context hint prefix (eg Label|lab).*
- [\\_mx](#) (...)  
*Get maxlength of a set of messages with a context hint prefix.*
- [\\_](#) (...)  
*Get a localized from the message catalog and deal with formating possible arguments, by calling ::msgcat::mc.*

### 8.10.1 Detailed Description

Localization functions.

Functions to fetch localized messages from the message catalog.

#### Author

Robert Heller <heller@deepsoft.com>

### 8.10.2 Package provided

gettext 1.0

### 8.10.3 Function Documentation

#### 8.10.3.1 `_()`

```
gettext::_ (
    ... )
```

Get a localized from the message catalog and deal with formating possible arguments, by calling `::msgcat::mc`.

Referenced by `FCFSupport::PauseCallback::Pause()`.

#### 8.10.3.2 `_m()`

```
gettext::_m (
    msgid ,
    ... )
```

Handle messages with a context hint prefix (eg `Label|lab`).

#### 8.10.3.3 `_mx()`

```
gettext::_mx (
    ... )
```

Get maxlength of a set of messages with a context hint prefix.

Used to compute label widths.

## 8.11 GRSupport Namespace Reference

Code to support the various graphics packages.

## Functions

- [\\_ROPI2](#) (name1, name2, op)  
*A variable trace to enforce the read-only-ness of PI2.*
- [\\_ROPI](#) (name1, name2, op)  
*A variable trace to enforce the read-only-ness of PI.*
- [DegreesToRadians](#) (degrees)  
*Function to convert from degrees to radians.*
- [RadiansToDegrees](#) (rads)  
*Function to convert from radians to degrees.*
- [VerifyDoubleMethod](#) ()  
*Snit macro defining a validate method for doubles.*
- [VerifyBooleanMethod](#) ()  
*Snit macro defining a validate method for booleans.*
- [VerifyIntegerMethod](#) ()  
*Snit macro defining a validate method integers.*
- [VerifyOrientationHVMMethod](#) ()  
*Snit macro defining a validate method for orientation (horizontal or vertical).*
- [VerifyColorMethod](#) ()  
*Snit macro defining a validate method for colors.*

## Variables

- [PI2](#)  
*Variable containing  $\pi/2$ .*
- [PI](#)  
*Variable containing  $\pi$ .*

### 8.11.1 Detailed Description

Code to support the various graphics packages.

Namespace where all of the graphics support code lives.

#### Author

Robert Heller <heller@deepsoft.com>

### 8.11.2 Package provided

grsupport 2.0

### 8.11.3 Function Documentation

### 8.11.3.1 `_ROPI()`

```
GRSupport::_ROPI (
    name1 ,
    name2 ,
    op ) [private]
```

A variable trace to enforce the read-only-ness of PI.

### 8.11.3.2 `_ROPI2()`

```
GRSupport::_ROPI2 (
    name1 ,
    name2 ,
    op ) [private]
```

A variable trace to enforce the read-only-ness of PI2.

### 8.11.3.3 `DegreesToRadians()`

```
GRSupport::DegreesToRadians (
    degrees )
```

Function to convert from degrees to radians.

#### Parameters

<i>degrees</i>	Value to convert to radians.
----------------	------------------------------

### 8.11.3.4 `RadiansToDegrees()`

```
GRSupport::RadiansToDegrees (
    rads )
```

Function to convert from radians to degrees.

#### Parameters

<i>rads</i>	Value to convert to degrees.
-------------	------------------------------

#### 8.11.3.5 VerifyBooleanMethod()

```
GRSupport::VerifyBooleanMethod ( )
```

Snit macro defining a validate method for booleans.

#### 8.11.3.6 VerifyColorMethod()

```
GRSupport::VerifyColorMethod ( )
```

Snit macro defining a validate method for colors.

#### 8.11.3.7 VerifyDoubleMethod()

```
GRSupport::VerifyDoubleMethod ( )
```

Snit macro defining a validate method for doubles.

#### 8.11.3.8 VerifyIntegerMethod()

```
GRSupport::VerifyIntegerMethod ( )
```

Snit macro defining a validate method integers.

#### 8.11.3.9 VerifyOrientationHVMethod()

```
GRSupport::VerifyOrientationHVMethod ( )
```

Snit macro defining a validate method for orientation (horizontal or vertical).

### 8.11.4 Variable Documentation

#### 8.11.4.1 PI

`GRSupport::PI`

Variable containing  $\pi$ .

This variable is read-only.

#### 8.11.4.2 PI2

`GRSupport::PI2`

Variable containing  $\pi/2$ .

This variable is read-only.

## 8.12 HTMLHelp Namespace Reference

[HTMLHelp](#) namespace, which contains the [HTMLHelp](#) snit widget adapter object and associated code.

### Classes

- class [HTMLHelp](#)

*A widget that implements a help dialog that renders HTML coded help pages (generally generated from LaTeX using `tex4ht's htlatex script`).*

### 8.12.1 Detailed Description

[HTMLHelp](#) namespace, which contains the [HTMLHelp](#) snit widget adapter object and associated code.

It uses code originally written by Stephen Uhler and modified by Clif Flynt (htmlib 0.3 through 0.3.4). I have modified it further and embedded into a snit widget adapter object to create a full featured help dialog object. I also added limited support for cascading style sheets.

#### Author

Stephen Uhler <stephen.uhler@sun.com>, Clif Flynt <clif@cflynt.com>, and Robert Heller <heller@deepsoft.com>.

### 8.12.2 Package provided

[HTMLHelp](#) 1.0



## 8.13 Instruments Namespace Reference

Namespace used for instruments code.

### Classes

- class [AnalogClock](#)  
*Analog clock instrument.*
- class [CabSignalLamp](#)  
*Cab signal lamp type.*
- class [DialInstrument](#)  
*Generic dial instrument.*
- class [DigitalClock](#)  
*Digital clock instrument.*
- class [DigitalInstrument](#)  
*Digital instrument.*

### Functions

- [CommonOptions](#) (defaultLabel)  
*Snit macro to define common options used by all instruments.*

#### 8.13.1 Detailed Description

Namespace used for instruments code.

#### Author

Robert Heller <heller@deepsoft.com>

#### 8.13.2 Package provided

[Instruments](#) 2.0

#### 8.13.3 Function Documentation

##### 8.13.3.1 CommonOptions()

```
Instruments::CommonOptions (  
    defaultLabel )
```

Snit macro to define common options used by all instruments.

## Parameters

<i>defaultLabel</i>	Default value for the label.
...	Options: <ul style="list-style-type: none"> <li>• -x The X coordinate of the instrument (default 0).</li> <li>• -y The Y coordinate of the instrument (default 0).</li> <li>• -size The size of the instrument (default 100).</li> <li>• -label The label of the instrument.</li> <li>• -labelcolor The color of the label (default black).</li> <li>• -labelfont The font of the label (default {Times 14 bold}).</li> <li>• -background The background color of the instrument (default blue).</li> <li>• -outline The outline color of the instrument (default black).</li> <li>• -scaleback The background color of the scale (default white).</li> <li>• -scaleticks The color of the scale ticks (default black).</li> <li>• -fontfamily The font family used on the instrument (default Courier).</li> </ul>

## 8.14 LabelComboBox Namespace Reference

This is a specialized form of the LabelFrame widget containing a ComboBox Widget.

### Functions

- [create](#) (path,...)  
*Procedure to create a [LabelComboBox](#).*
- [configure](#) (path,...)  
*Procedure to configure a [LabelComboBox](#).*
- [cget](#) (path, option)  
*Procedure to get a configuration option.*
- [bind](#) (path,...)  
*Procedure to set a binding on the ComboBox entry.*
- [get](#) (path,...)  
*Procedure to get the ComboBox value.*
- [getlistbox](#) (path,...)  
*Procedure to get the listbox of the ComboBox widget.*
- [getvalue](#) (path,...)  
*Procedure to get the value of the ComboBox.*
- [icursor](#) (path,...)  
*Pass through procedure for the ComboBox icursor function.*

- [post](#) (path,...)  
*Pass through procedure for the ComboBox post function.*
- [setvalue](#) (path,...)  
*Pass through procedure for the ComboBox setvalue function.*
- [unpost](#) (path,...)  
*Pass through procedure for the ComboBox unpost function.*
- [\\_path\\_command](#) (path, cmd, larg)  
*Path command for this megawidget.*
- [\\_destroy](#) (path)  
*Destructor function.*

### 8.14.1 Detailed Description

This is a specialized form of the LabelFrame widget containing a ComboBox Widget.

Most of the resources from the LabelFrame and ComboBox widgets are included in this widget.

#### Parameters

<i>path</i>	The widget path.
...	Options: <ul style="list-style-type: none"> <li>• -labeljustify From LabelFrame (-justify).</li> <li>• -labelwidth From LabelFrame (-width).</li> <li>• -labelanchor From LabelFrame (-anchor).</li> <li>• -labelheight From LabelFrame (-height).</li> <li>• -labelfont From LabelFrame (-font).</li> <li>• -labeltextvariable From LabelFrame (-textvariable).</li> <li>• -label From LabelFrame (-text).</li> <li>• -comboboxfg From ComboBox (-foreground).</li> <li>• -comboboxbg From ComboBox (-background).</li> <li>• -comboboxheight From ComboBox (-height).</li> <li>• -comboboxlistboxwidth From ComboBox (-listboxwidth).</li> <li>• -values From ComboBox.</li> </ul>

#### Author

Robert Heller <heller@deepsoft.com>

### 8.14.2 Package provided

BWLabelComboBox 1.0

### 8.14.3 Function Documentation

#### 8.14.3.1 `_destroy()`

```
LabelComboBox::_destroy (
    path ) [private]
```

Destructor function.

##### Parameters

<i>path</i>	The path of the megawidget.
-------------	-----------------------------

#### 8.14.3.2 `_path_command()`

```
LabelComboBox::_path_command (
    path ,
    cmd ,
    larg ) [private]
```

Path command for this megawidget.

Implements all of the megawidget commands.

##### Parameters

<i>path</i>	The path of the megawidget.
<i>cmd</i>	The command name.
<i>larg</i>	The command argument.

#### 8.14.3.3 `bind()`

```
LabelComboBox::bind (
    path ,
    ... )
```

Procedure to set a binding on the ComboBox entry.

## Parameters

<i>path</i>	Path to the new widget.
...	Arguments to pass to the ComboBox bind procedure.

## 8.14.3.4 cget()

```
LabelComboBox::cget (
    path ,
    option )
```

Procedure to get a configuration option.

## Parameters

<i>path</i>	Path to the new widget.
<i>option</i>	Configuration option to get.

## 8.14.3.5 configure()

```
LabelComboBox::configure (
    path ,
    ... )
```

Procedure to configure a [LabelComboBox](#).

## Parameters

<i>path</i>	Path to the new widget.
...	Configuration options.

## 8.14.3.6 create()

```
LabelComboBox::create (
    path ,
    ... )
```

Procedure to create a [LabelComboBox](#).

**Parameters**

<i>path</i>	Path to the new widget.
...	Configuration options.

**8.14.3.7 get()**

```
LabelComboBox::get (
    path ,
    ... )
```

Procedure to get the ComboBox value.

**Parameters**

<i>path</i>	Path to the new widget.
...	Arguments to pass to the ComboBox get procedure

**8.14.3.8 getlistbox()**

```
LabelComboBox::getlistbox (
    path ,
    ... )
```

Procedure to get the listbox of the ComboBox widget.

**Parameters**

<i>path</i>	Path to the new widget.
...	Arguments to pass to the ComboBox getlistbox procedure.

**8.14.3.9 getvalue()**

```
LabelComboBox::getvalue (
    path ,
    ... )
```

Procedure to get the value of the ComboBox.

## Parameters

<i>path</i>	Path to the new widget.
...	Arguments to pass to the ComboBox getvalue procedure.

## 8.14.3.10 icursor()

```
LabelComboBox::icursor (
    path ,
    ... )
```

Pass through procedure for the ComboBox icursor function.

## Parameters

<i>path</i>	Path to the new widget.
...	Arguments to pass to the ComboBox icursor function.

## 8.14.3.11 post()

```
LabelComboBox::post (
    path ,
    ... )
```

Pass through procedure for the ComboBox post function.

## Parameters

<i>path</i>	Path to the new widget.
...	Arguments to pass to the ComboBox post function.

## 8.14.3.12 setvalue()

```
LabelComboBox::setvalue (
    path ,
    ... )
```

Pass through procedure for the ComboBox setvalue function.

**Parameters**

<i>path</i>	Path to the new widget.
...	Arguments to pass to the ComboBox setValue function.

**8.14.3.13 unpost()**

```
LabelComboBox::unpost (
    path ,
    ... )
```

Pass through procedure for the ComboBox unpost function.

**Parameters**

<i>path</i>	Path to the new widget.
...	Arguments to pass to the ComboBox unpost function.

**8.15 LabelSelectColor Namespace Reference**

This package provides a BWidget style megawidget for selecting colors, in the same style as a LabelEntry widget.

**Functions**

- [create](#) (path,...)  
*Creation procedure.*
- [ColorPopup](#) (path)  
*Procedure bound to the palette button to select a color.*
- [configure](#) (path,...)  
*Configuration procedure: configure one or more options for this widget.*
- [cget](#) (path, option)  
*Configuration option accessor procedure: access one option directly.*
- [\\_path\\_command](#) (path, cmd, larg)  
*Path command for this megawidget.*
- [\\_destroy](#) (path)  
*Destructor function.*

**8.15.1 Detailed Description**

This package provides a BWidget style megawidget for selecting colors, in the same style as a LabelEntry widget.

This megawidget includes resources from the LabelFrame, Entry, and Button widgets.



## Parameters

<i>pathname</i>	Widget pathname.
...	Options: <ul style="list-style-type: none"> <li>• <code>-labeljustify</code> From <code>LabelFrame</code> (<code>-justify</code>).</li> <li>• <code>-labelwidth</code> From <code>LabelFrame</code> (<code>-width</code>).</li> <li>• <code>-labelanchor</code> From <code>LabelFrame</code> (<code>-anchor</code>).</li> <li>• <code>-labelheight</code> From <code>LabelFrame</code> (<code>-height</code>).</li> <li>• <code>-labelfont</code> From <code>LabelFrame</code> (<code>-font</code>).</li> <li>• <code>-labeltextvariable</code> From <code>LabelFrame</code> (<code>-textvariable</code>).</li> <li>• <code>-label</code> From <code>LabelFrame</code> (<code>-text</code>).</li> <li>• <code>-selectcolorfg</code> From <code>Entry</code> (<code>-foreground</code>).</li> <li>• <code>-selectcolorbg</code> From <code>Entry</code> (<code>-background</code>).</li> <li>• <code>-text</code> From <code>Entry</code>.</li> <li>• <code>-buttonfg</code> From <code>Button</code> (<code>-foreground</code>).</li> <li>• <code>-buttonbg</code> From <code>Button</code> (<code>-background</code>).</li> <li>• <code>-buttonactivebg</code> From <code>Button</code> (<code>-activebackground</code>).</li> <li>• <code>-buttonactivefg</code> From <code>Button</code> (<code>-activeforeground</code>).</li> <li>• <code>-buttondisabledfg</code> From <code>Button</code> (<code>-disabledforeground</code>).</li> <li>• <code>-buttonhighlightbg</code> From <code>Button</code> (<code>-highlightbackground</code>).</li> <li>• <code>-buttonhighlightcolor</code> From <code>Button</code> (<code>-highlightcolor</code>).</li> </ul>

## Author

Robert Heller <heller@deepsoft.com>

## 8.15.2 Package provided

[LabelSelectColor](#) 1.0

## 8.15.3 Function Documentation

8.15.3.1 `_destroy()`

```
LabelSelectColor::_destroy (
    path ) [private]
```

Destructor function.

**Parameters**

<i>path</i>	– The path of the megawidget.
-------------	-------------------------------

**8.15.3.2 `_path_command()`**

```
LabelSelectColor::_path_command (  
    path ,  
    cmd ,  
    larg ) [private]
```

Path command for this megawidget.

Implements all of the megawidget commands.

**Parameters**

<i>path</i>	– The path of the megawidget.
<i>cmd</i>	– The command name.
<i>larg</i>	– The command argument.

**8.15.3.3 `cget()`**

```
LabelSelectColor::cget (  
    path ,  
    option )
```

Configuration option accessor procedure: access one option directly.

**Parameters**

<i>path</i>	– The path of the megawidget.
<i>option</i>	– The option to access

**8.15.3.4 `ColorPopup()`**

```
LabelSelectColor::ColorPopup (  
    path )
```

Procedure bound to the palette button to select a color.

**Parameters**

<i>path</i>	– The path of the megawidget.
-------------	-------------------------------

**8.15.3.5 configure()**

```
LabelSelectColor::configure (
    path ,
    ... )
```

Configuration procedure: configure one or more options for this widget.

**Parameters**

<i>path</i>	– The path of the megawidget.
...	– Option value pairs.

**8.15.3.6 create()**

```
LabelSelectColor::create (
    path ,
    ... )
```

Creation procedure.

**Parameters**

<i>path</i>	– The megawidget's path.
...	– Options for this widget.

**8.16 LabelSpinBox Namespace Reference**

This is a specialized form of the LabelFrame widget containing a SpinBox Widget.

**Functions**

- [create](#) (path,...)

*Procedure to create a [LabelSpinBox](#).*

- [configure](#) (path,...)

*Procedure to configure a [LabelSpinBox](#).*

- [cget](#) (path, option)

*Procedure to get a configuration option.*

- [setvalue](#) (path,...)

*Procedure to set the value of the SpinBox.*

- [getvalue](#) (path,...)

*Procedure to get the value of the SpinBox.*

- [bind](#) (path,...)

*Procedure to set a binding on the SpinBox entry.*

- [\\_path\\_command](#) (path, cmd, larg)

*Path command for this megawidget.*

- [\\_destroy](#) (path)

*Destructor function.*

### 8.16.1 Detailed Description

This is a specialized form of the LabelFrame widget containing a SpinBox Widget.

Most of the resources from the LabelFrame and SpinBox widgets are included in this widget.

#### Parameters

<i>path</i>	The widget path
...	Options: <ul style="list-style-type: none"> <li>• -labeljustify From LabelFrame (-justify).</li> <li>• -labelwidth From LabelFrame (-width).</li> <li>• -labelanchor From LabelFrame (-anchor).</li> <li>• -labelheight From LabelFrame (-height).</li> <li>• -labelfont From LabelFrame (-font).</li> <li>• -labeltextvariable From LabelFrame (-textvariable).</li> <li>• -label From LabelFrame (-text).</li> <li>• -spinboxfg From SpinBox (-foreground).</li> <li>• -spinboxbg From SpinBox (-background).</li> <li>• -range From SpinBox.</li> <li>• -values From SpinBox.</li> </ul>

**Author**

Robert Heller <heller@deepsoft.com>

**8.16.2 Package provided**

BWLabelSpinBox 1.0

**8.16.3 Function Documentation****8.16.3.1 \_destroy()**

```
LabelSpinBox::_destroy (  
    path ) [private]
```

Destructor function.

**Parameters**

<i>path</i>	The path of the megawidget.
-------------	-----------------------------

**8.16.3.2 \_path\_command()**

```
LabelSpinBox::_path_command (  
    path ,  
    cmd ,  
    larg ) [private]
```

Path command for this megawidget.

Implements all of the megawidget commands.

**Parameters**

<i>path</i>	The path of the megawidget.
<i>cmd</i>	The command name.
<i>larg</i>	The command argument.

### 8.16.3.3 bind()

```
LabelSpinBox::bind (
    path ,
    ... )
```

Procedure to set a binding on the SpinBox entry.

#### Parameters

<i>path</i>	Path to the new widget.
<i>...</i>	Arguments to pass to the SpinBox bind procedure.

### 8.16.3.4 cget()

```
LabelSpinBox::cget (
    path ,
    option )
```

Procedure to get a configuration option.

#### Parameters

<i>path</i>	Path to the new widget.
<i>option</i>	Configuration option to get.

### 8.16.3.5 configure()

```
LabelSpinBox::configure (
    path ,
    ... )
```

Procedure to configure a [LabelSpinBox](#).

#### Parameters

<i>path</i>	Path to the new widget.
<i>...</i>	Configuration options.

#### 8.16.3.6 create()

```
LabelSpinBox::create (
    path ,
    ... )
```

Procedure to create a [LabelSpinBox](#).

##### Parameters

<i>path</i>	Path to the new widget.
...	Configuration options.

#### 8.16.3.7 getvalue()

```
LabelSpinBox::getvalue (
    path ,
    ... )
```

Procedure to get the value of the SpinBox.

##### Parameters

<i>path</i>	Path to the new widget.
...	Arguments to pass to the SpinBox getvalue procedure.

#### 8.16.3.8 setvalue()

```
LabelSpinBox::setvalue (
    path ,
    ... )
```

Procedure to set the value of the SpinBox.

##### Parameters

<i>path</i>	Path to the new widget.
...	Arguments to pass to the SpinBox setvalue procedure.



## 8.17 LCARS Namespace Reference

Namespace where the [LCARS](#) code lives.

### 8.17.1 Detailed Description

Namespace where the [LCARS](#) code lives.

#### Author

Robert Heller <heller@deepsoft.com>

### 8.17.2 Package provided

LCARSWidgets 2.0

## 8.18 lcc Namespace Reference

Namespace that holds the LCC interface code.

### Classes

- class [CanAlias](#)  
*Implements a CAN Alias.*
- class [CANGridConnect](#)  
*Base class to connect to a CAN bus using GridConnect formatted message over.*
- class [CANGridConnectOverCANSocket](#)  
*Connect to a CAN bus using GridConnect formatted message over a CAN Socket connection.*
- class [CANGridConnectOverTcp](#)  
*Connect to a CAN bus using GridConnect formatted message over a Tcp/Ip connection.*
- class [CANGridConnectOverUSBSerial](#)  
*Connect to a CAN bus using GridConnect formatted message over a USB Serial port.*
- class [CANHeader](#)  
*CAN Header type.*
- class [CanMessage](#)  
*A CAN Message, containing a 29-bit header and upto 8 bytes of data.*
- class [CanTransport](#)  
*Logical transport of CAN Messages.*
- class [ConfigMemory](#)  
*Configure memory.*
- class [ConfigOptions](#)  
*Display memory config options.*

- class [ConfigurationEditor](#)  
*Generate OpenLCB Memory Configuration Window.*
- class [EventID](#)  
*An event id structure.*
- class [EventID\\_or\\_null](#)  
*An [EventID](#) or empty string.*
- class [EventLog](#)  
*Event received log, with event sender.*
- class [EventReceived](#)  
*Display a received event.*
- class [GridConnectMessage](#)  
*A Grid Connect formatted CAN message.*
- class [GridConnectReply](#)  
*A Grid Connect formatted CAN message (reply).*
- class [MTIDetail](#)  
*MTI Header type, detailed version.*
- class [MTIHeader](#)  
*MTI Header type.*
- class [nid\\_or\\_null](#)  
*Node ID regexp pattern or the empty string.*
- class [OpenLCBMessage](#)  
*OpenLCB Message type.*
- class [OpenLCBNode](#)  
*Connect to a OpenLCB interface.*
- class [OpenLCBOverTcp](#)  
*Connect to a OpenLCB over Tcp/Ip.*
- class [OpenLCBProtocols](#)  
*Supported LCC Protocol name type.*
- class [SendEvent](#)  
*Send Event Dialog – send PCRE message.*

## Typedefs

- typedef int [twobits](#)  
*A 2 bit integer.*
- typedef int [threebits](#)  
*A 3 bit integer.*
- typedef int [fivebits](#)  
*A 5 bit integer.*
- typedef int [sixbits](#)  
*A 6 bit integer.*
- typedef int [length](#)  
*An integer from 1 to 64.*
- typedef int [byte](#)  
*An 8-bit unsigned byte.*
- typedef int [twelvebits](#)

- A 12 bit integer.*
  - typedef int [fifteenbits](#)
- A 15 bit integer.*
  - typedef int [sixteenbits](#)
- A 16 bit integer.*
  - typedef int [headerword](#)
- A 29 bit integer.*
  - typedef int [uint32](#)
- A 32 bit unsigned integer.*
  - typedef listtype [eightbytes](#)
- A list of bytes, from 0 to 8 elements.*
  - typedef listtype [bytelist72](#)
- A list of bytes, from 0 to 72 elements.*
  - typedef listtype [bytelist](#)
- A list of bytes, unbounded.*
  - typedef listtype [databuf](#)
- A list of bytes, from 1 to 64 elements.*
  - typedef char \* [nid](#)
- Node ID regexp pattern.*

## Enumerations

- enum [datagramcontent](#) {  
[complete](#), [first](#), [middle](#), [last](#),  
[stream](#) }*Datagram and stream types.*
- enum [eventvalidity](#) { [valid](#), [invalid](#), [unknown](#) }*Event validity.*

## Functions

- [AbstractMessage](#) ()*Define common variables and accessor methods.*
- [AbstractMRMessage](#) ()*Macro to create common methods and variables for an AbstractMRMessage*

### 8.18.1 Detailed Description

Namespace that holds the LCC interface code.

This is a cross-platform implementation ...

#### Author

Robert Heller <heller@deepsoft.com>

### 8.18.2 Package provided

LCC 1.0

### 8.18.3 Package provided

ConfigDialogs 1.0

### 8.18.4 Package provided

EventDialogs 1.0

### 8.18.5 Package provided

[ConfigurationEditor](#) 1.0

### 8.18.6 Typedef Documentation

#### 8.18.6.1 byte

unsigned char [lcc::byte](#)

An 8-bit unsigned byte.

#### 8.18.6.2 bytelist

list [lcc::bytelist](#)

A list of bytes, unbounded.

#### 8.18.6.3 bytelist72

list [lcc::bytelist72](#)

A list of bytes, from 0 to 72 elements.

#### 8.18.6.4 databuf

```
list lcc::databuf
```

A list of bytes, from 1 to 64 elements.

#### 8.18.6.5 eightbytes

```
list lcc::eightbytes
```

A list of bytes, from 0 to 8 elements.

#### 8.18.6.6 fifteenbits

```
int lcc::fifteenbits
```

A 15 bit integer.

#### 8.18.6.7 fivebits

```
int lcc::fivebits
```

A 5 bit integer.

#### 8.18.6.8 headerword

```
int lcc::headerword
```

A 29 bit integer.

#### 8.18.6.9 length

```
int lcc::length
```

An integer from 1 to 64.

**8.18.6.10 nid**

```
string lcc::nid
```

Node ID regexp pattern.

A Node Id is six bytes as pairs of hex digits separated by colons (:).

**8.18.6.11 sixbits**

```
int lcc::sixbits
```

A 6 bit integer.

**8.18.6.12 sixteenbits**

```
int lcc::sixteenbits
```

A 16 bit integer.

**8.18.6.13 threebits**

```
int lcc::threebits
```

A 3 bit integer.

**8.18.6.14 twelvebits**

```
int lcc::twelvebits
```

A 12 bit integer.

**8.18.6.15 twobits**

```
int lcc::twobits
```

A 2 bit integer.

#### 8.18.6.16 uint32

```
int lcc::uint32
```

A 32 bit unsigned integer.

### 8.18.7 Enumeration Type Documentation

#### 8.18.7.1 datagramcontent

```
enum lcc::datagramcontent
```

Datagram and stream types.

##### Enumerator

complete	One frame datagram.
first	First frame datagram.
middle	Middle frame datagram.
last	Last frame datagram.
stream	Stream frame.

#### 8.18.7.2 eventvalidity

```
enum lcc::eventvalidity
```

Event validity.

##### Enumerator

valid	Currently valid.
invalid	Currently invalid.
unknown	Currently unknown.

### 8.18.8 Function Documentation

#### 8.18.8.1 AbstractMessage()

```
lcc::AbstractMessage ( )
```

Define common variables and accessor methods.

#### 8.18.8.2 AbstractMRMessage()

```
lcc::AbstractMRMessage ( )
```

Macro to create common methods and variables for an AbstractMRMessage

## 8.19 linuxgpio Namespace Reference

Linux GPIO Interface.

### Classes

- class [GpioInputActiveHigh](#)  
*Input pin, active high (high is true).*
- class [GpioInputActiveLow](#)  
*Input pin, active low (low is true).*
- class [GpioOutputSafeHigh](#)  
*Output pin, initialized to high.*
- class [GpioOutputSafeHighInvert](#)  
*Output pin, initialized to high, inverted.*
- class [GpioOutputSafeLow](#)  
*Output pin, initialized to low.*
- class [GpioOutputSafeLowInverted](#)  
*Output pin, initialized to low, with inverted logic.*
- class [LinuxGpio](#)  
*Base generic GPIO interface class.*

### Typedefs

- typedef int [pinnotype](#)  
*Pin number type, a positive integer.*

### Enumerations

- enum [pindirection](#) { [in](#), [out](#), [high](#), [low](#) }  
*Pin direction and initial type code.*



### 8.19.1 Detailed Description

Linux GPIO Interface.

This is the portable implementation of GPIO under Linux, using the sysfs file system (/sys/class/gpio/...). This code should work on all SBC / development boards that run Linux (Raspberry Pis, Beagle Bones, Banana Pis, etc.).

Author

Robert Heller <heller@deepsoft.com>

### 8.19.2 Package provided

[LinuxGpio](#) 1.0.0

### 8.19.3 Typedef Documentation

#### 8.19.3.1 pinnotype

```
typedef int linuxgpio::pinnotype
```

Pin number type, a positive integer.

### 8.19.4 Enumeration Type Documentation

#### 8.19.4.1 pindirection

```
enum linuxgpio::pindirection
```

Pin direction and initial type code.

Enumerator

in	Input Pin.
out	Output Pin.
high	Output Pin, initialized to high.
low	Output Pin, initialized to low.

## 8.20 nce Namespace Reference

Namespace that holds the [NCE](#) interface code.

### Classes

- class [NCE](#)  
*Main [NCE](#) Cab Bus interface class.*

### Typedefs

- typedef int [LocoAddress](#)  
*Locomotive address type.*
- typedef int [ConsistAddress](#)  
*Consist address type.*
- typedef int [AccessoryNumber](#)  
*Accessory address type.*
- typedef int [MacroNumber](#)  
*[NCE](#) Macro number.*
- typedef int [CabNumber](#)  
*Cab number type.*
- typedef int [Hours](#)  
*Hours type.*
- typedef int [Minutes](#)  
*Minutes type.*
- typedef int [ScaleClockRatio](#)  
*Scale clock ratio range.*
- typedef int [EchoMode](#)  
*This is really should be an enumeration, but works as a limited range integer.*
- typedef int [Speed28](#)  
*28 speed step speeds.*
- typedef int [Speed128](#)  
*128 speed step speeds.*
- typedef int [CSAddress](#)  
*CSAddress type.*
- typedef int [UByte](#)  
*Unsigned byte type (data byte).*
- typedef listtype [RAMData](#)  
*Datalist for RAM data, 1 to 16 unsigned bytes.*
- typedef char \* [LCDMessage16](#)  
*Data for 16 character LCD lines.*
- typedef char \* [LCDMessage8](#)  
*Data for 8 character LCD lines.*
- typedef listtype [RawPacket](#)  
*Raw packets for writing raw packets to the temp queue.*

- typedef listtype [RawTrackPacket](#)  
*Raw packets for writing raw packets to the track queue.*
- typedef listtype [RAMData8](#)  
*Datalist for RAM data 8 unsigned bytes.*
- typedef int [MomentumLevel](#)  
*Momentum level.*
- typedef int [AspectBits](#)  
*Aspect bit mask.*

## Enumerations

- enum [SpeedMode](#) { [S14](#), [S28](#), [S128](#) }
- enum [Direction](#) { [Forward](#), [Reverse](#) }

## Functions

- [ErrorMessage](#) (code)  
*Return the error message, given the error code.*

### 8.20.1 Detailed Description

Namespace that holds the [NCE](#) interface code.

This is a cross-platform implementation the [NCE](#) Cab Bus serial port interface. Based on documentation provided by [NCE](#) ([usb\\_1.pdf](#) and [Bincmds.pdf](#)).

Basically, the way this code works is to use a class to interface to the real RS232 port attached to a CS02 command station OR the 'virtual' serial port implemented by the [NCE](#) USB Interface Board connected to the [NCE](#) Cab Bus.

#### Author

Robert Heller <[heller@deepsoft.com](mailto:heller@deepsoft.com)>

### 8.20.2 Package provided

[NCE](#) 1.0.0

### 8.20.3 Typedef Documentation

#### 8.20.3.1 AccessoryNumber

```
int nce::AccessoryNumber
```

Accessory address type.

#### 8.20.3.2 AspectBits

```
int nce::AspectBits
```

Aspect bit mask.

#### 8.20.3.3 CabNumber

```
int nce::CabNumber
```

Cab number type.

#### 8.20.3.4 ConsistAddress

```
int nce::ConsistAddress
```

Consist address type.

#### 8.20.3.5 CSAddress

```
unsigned short int nce::CSAddress
```

CSAddress type.

#### 8.20.3.6 EchoMode

```
int nce::EchoMode
```

This is really should be an enumeration, but works as a limited range integer.

Allowed values are:

- 0 No echo.
- 1 Echo 1st byte of command.
- 2 Echo entire command.

#### 8.20.3.7 Hours

```
int nce::Hours
```

Hours type.

#### 8.20.3.8 LCDMessage16

```
char nce::LCDMessage16[16]
```

Data for 16 character LCD lines.

#### 8.20.3.9 LCDMessage8

```
char nce::LCDMessage8[8]
```

Data for 8 character LCD lines.

#### 8.20.3.10 LocoAddress

```
int nce::LocoAddress
```

Locomotive address type.

#### 8.20.3.11 MacroNumber

int `nce::MacroNumber`

**NCE** Macro number.

#### 8.20.3.12 Minutes

int `nce::Minutes`

Minutes type.

#### 8.20.3.13 MomentumLevel

int `nce::MomentumLevel`

Momentum level.

#### 8.20.3.14 RAMData

list `nce::RAMData`

Datalist for RAM data, 1 to 16 unsigned bytes.

#### 8.20.3.15 RAMData8

list `nce::RAMData8`

Datalist for RAM data 8 unsigned bytes.

#### 8.20.3.16 RawPacket

list `nce::RawPacket`

Raw packets for writing raw packets to the temp queue.

#### 8.20.3.17 RawTrackPacket

```
list nce::RawTrackPacket
```

Raw packets for writing raw packets to the track queue.

#### 8.20.3.18 ScaleClockRatio

```
int nce::ScaleClockRatio
```

Scale clock ratio range.

#### 8.20.3.19 Speed128

```
int nce::Speed128
```

128 speed step speeds.

#### 8.20.3.20 Speed28

```
int nce::Speed28
```

28 speed step speeds.

#### 8.20.3.21 UByte

```
unsigned char nce::UByte
```

Unsigned byte type (data byte).

### 8.20.4 Enumeration Type Documentation

#### 8.20.4.1 Direction

```
enum nce::Direction
```

**Enumerator**

Forward	Forward direction.
Reverse	Reverse direction.

**8.20.4.2 SpeedMode**

enum `nce::SpeedMode`

**Enumerator**

S14	Fourteen speed step mode.
S28	Twenty eight speed step mode.
S128	128 speed step mode.

**8.20.5 Function Documentation****8.20.5.1 ErrorMessage()**

```
nce::ErrorMessage (
    code )
```

Return the error message, given the error code.

This function returns the error message associated with a given error code.

**Parameters**

<code>code</code>	Error code returned.
-------------------	----------------------

**Returns**

A localized error message string.

**Author**

Robert Heller <heller@deepsoft.com>



## 8.21 OvalWidgets Namespace Reference

These oval shaped widgets are much like the Star Trek NG computer screens.

### Classes

- class [OvalButton](#)  
*Oval button.*
- class [OvalRoundCornerRectangle](#)  
*Oval Round Corner Rectangle.*
- class [OvalScale](#)  
*An oval scale widget, much like a standard Tk scale widget.*
- class [OvalSlider](#)  
*Oval Slider.*
- class [OvalScrollBar](#)  
*Oval ScrollBar.*

### Functions

- [XYWH](#) (width, height)  
*Defines the options for position (-x,-y) and size (-width,-height).*
- [ColorOptionMethods](#) ()  
*Snit macro to default color option methods.*
- [CommonValidateMethods](#) ()  
*Macro to include the common validation methods.*
- [ColorFillOption](#) (optspec, default)  
*Method to define a fill color option.*
- [ColorOutlineOption](#) (optspec, default)  
*Method to define an outline color option.*
- [FontFamily](#) (default)  
*Macro to define the -fontfamily option.*
- [SquareEndOptions](#) ()  
*Macro to define the square end options (-rightsquare, -leftsquare).*
- [\\_VerifyFont](#) (option, value)  
*Method to validate a font value.*
- [\\_ConfigureFont](#) (option, value)  
*Method to configure a font value.*
- [\\_ConfigureText](#) (option, value)  
*Method to configure the text of the button.*
- [\\_VerifyIntegerOrEmpty](#) (option, value)  
*Method to validate an integer or empty string option.*
- [OvalLabel](#) (name, \_canvas,...)  
*Construct some text.*
- [~OvalLabel](#) ()  
*Destructor free up all resources.*
- [\\_UnderSplit](#) (beforevar, undervar, aftervar)  
*Method to split label text into before, under, and after segments.*

## Variables

- [HBar](#)  
*Holds the horizontal bar bitmap.*
- [VBar](#)  
*Holds the vertical bar bitmap.*
- var [canvas](#)  
*Canvas the widget is on.*

### 8.21.1 Detailed Description

These oval shaped widgets are much like the Star Trek NG computer screens.

#### Author

Robert Heller <heller@deepsoft.com>

### 8.21.2 Package provided

[OvalWidgets](#) 2.0

### 8.21.3 Function Documentation

#### 8.21.3.1 \_ConfigureFont()

```
OvalWidgets::_ConfigureFont (
    option ,
    value ) [private]
```

Method to configure a font value.

#### Parameters

<i>option</i>	The name of the option to configure.
<i>value</i>	The value of the option.

#### 8.21.3.2 \_ConfigureText()

```
OvalWidgets::_ConfigureText (
```

```
option ,  
value ) [private]
```

Method to configure the text of the button.

#### Parameters

<i>option</i>	The name of the option to configure.
<i>value</i>	The value to configure it to.

#### 8.21.3.3 \_UnderSplit()

```
OvalWidgets::_UnderSplit (  
    beforevar ,  
    undervar ,  
    aftervar ) [private]
```

Method to split label text into before, under, and after segments.

#### Parameters

<i>beforevar</i>	The name of the before variable.
<i>undervar</i>	The name of the under variable.
<i>aftervar</i>	The name of the after variable.

#### 8.21.3.4 \_VerifyFont()

```
OvalWidgets::_VerifyFont (  
    option ,  
    value ) [private]
```

Method to validate a font value.

#### Parameters

<i>option</i>	The name of the option to validate.
<i>value</i>	The value of the option.

#### 8.21.3.5 \_VerifyIntegerOrEmpty()

```
OvalWidgets::_VerifyIntegerOrEmpty (
    option ,
    value ) [private]
```

Method to validate an integer or empty string option.

##### Parameters

<i>option</i>	The name of the option to validate.
<i>value</i>	The value of the option.

#### 8.21.3.6 ColorFillOption()

```
OvalWidgets::ColorFillOption (
    optspec ,
    default )
```

Method to define a fill color option.

##### Parameters

<i>optspec</i>	The option specification
<i>default</i>	The default value.

#### 8.21.3.7 ColorOptionMethods()

```
OvalWidgets::ColorOptionMethods ( )
```

Snit macro to default color option methods.

#### 8.21.3.8 ColorOutlineOption()

```
OvalWidgets::ColorOutlineOption (
    optspec ,
    default )
```

Method to define an outline color option.

## Parameters

<i>optspec</i>	The option specification
<i>default</i>	The default value.

**8.21.3.9 CommonValidateMethods()**

```
OvalWidgets::CommonValidateMethods ( )
```

Macro to include the common validation methods.

**8.21.3.10 FontFamily()**

```
OvalWidgets::FontFamily (
    default )
```

Macro to define the -fontfamily option.

## Parameters

<i>default</i>	The default font family.
----------------	--------------------------

**8.21.3.11 OvalLabel()**

```
OvalWidgets::OvalLabel (
    name ,
    _canvas ,
    ... )
```

Construct some text.

## Parameters

<i>_canvas</i>	The canvas to draw the text on.
...	The option value list.

### 8.21.3.12 SquareEndOptions()

```
OvalWidgets::SquareEndOptions ( )
```

Macro to define the square end options (-rightsquare, -leftsquare).

### 8.21.3.13 XYWH()

```
OvalWidgets::XYWH (
    width ,
    height )
```

Defines the options for position (-x,-y) and size (-width,-height).

#### Parameters

<i>width</i>	Default width.
<i>height</i>	Default height.

### 8.21.3.14 ~OvalLabel()

```
OvalWidgets::~~OvalLabel ( )
```

Destructor free up all resources.

## 8.21.4 Variable Documentation

### 8.21.4.1 canvas

```
var OvalWidgets::canvas [private]
```

#### Initial value:

```
{
public:
```

```
_ConfigureXY ( option, value)
```

Canvas the widget is on.

#### 8.21.4.2 HBar

OvalWidgets::HBar

Holds the horizontal bar bitmap.

#### 8.21.4.3 VBar

OvalWidgets::VBar

Holds the vertical bar bitmap.

## 8.22 PanedWindow Namespace Reference

A modified version of the BWidget [PanedWindow](#).

### 8.22.1 Detailed Description

A modified version of the BWidget [PanedWindow](#).

Added methods: index, paneconfigure, panecget, Pane::configure, and Pane::cget. Added a -name option to the add function. The modification allow for accessing and configuring the panes added to the [PanedWindow](#), mostly to get size information. This makes the BWidget [PanedWindow](#) more like the Tix Paned Window widget. (See the BWidget documentation for [PanedWindow](#).)

#### Author

Robert Heller <heller@deepsoft.com>

### 8.22.2 Package provided

DWpanedw 1.0

## 8.23 Parsers Namespace Reference

File-based parser classes.

## Classes

- class [BezierBody](#)  
*List of Bezier body lines (T, E, S, and C lines).*
- class [BezierBodyElt](#)  
*Bezier Body elements: T, E, S, and C lines are collected.*
- class [CornuBody](#)  
*List of Cornu body lines (T, E, S, and C lines).*
- class [CornuBodyElt](#)  
*Cornu Body elements: T, E, S, and C lines are collected.*
- class [IntegerList](#)  
*The [IntegerList](#) class implements a linked list of integers, used for turnout route lists.*
- class [LayoutFile](#)  
*File to parse an XTrkCad layout file and create a track graph.*
- class [MRRXtrkCad](#)  
*[MRRXtrkCad](#) parser class.*
- class [ParseFile](#)  
*Virtual base class for file-based parsers.*
- struct [RouteVec](#)  
*Route structure.*
- struct [SegPos](#)  
*Segment position, endpoint or other coordinate.*
- struct [SegVector](#)  
*Segemnt structure.*
- class [TrackBody](#)  
*List of track endpoints (T and E lines).*
- class [TrackBodyElt](#)  
*Track endpoint elements (T and E lines).*
- class [TrackGraph](#)  
*Track Graph class, which encapsulates the track graphs.*
- class [TurnoutBody](#)  
*List of turnout body lines (T, E, P, S, C, and J lines).*
- class [TurnoutBodyElt](#)  
*Turnout body elements: T, E, P, S, C, and J lines are collected.*
- struct [TurnoutGraphic](#)  
*Structure holding a turnout's graphical information.*
- struct [TurnoutRoutelist](#)  
*Turnout route list structure.*

### 8.23.1 Detailed Description

File-based parser classes.

These are file-based parser classes. Right now only one parser for XTrkCAD layout files. Other classes might be added later.



Included are classes used by the XTrkCAD parser. These classes are used to store the track plan information in an XTrkCAD layout file, specifically as it relates to operating issues, such as dispatching and signaling.

The track plan is loaded into a directed graph representation, where each node is one logical piece of trackwork. From this graph representation a schematic display could be created in a semi-automated way.

This package features the use of the Boost Graph Library as the underlying structure for the track graph built from reading in XTrkCAD layout files.

#### Author

Robert Heller <heller@deepsoft.com>

### 8.23.2 Tcl Package Provided

Mrr 2.2.1

### 8.23.3 Library Provided

libMRRParserClasses 2.2.1

## 8.24 raildriver Namespace Reference

Namespace that holds the Raildriver Client class code.

### Classes

- class [RaildriverClient](#)  
*Raildriver Client class – connects to the Raildriver daemon.*

### Typedefs

- typedef listtype [eventlist](#)  
*List of event codes.*

### Enumerations

- enum [RaildriverEvents](#) {  
    [REVERSER](#), [THROTTLE](#), [AUTOBRAKE](#), [INDEPENDBRK](#),  
    [BAILOFF](#), [WIPER](#), [HEADLIGHT](#), [DIGITAL1](#),  
    [DIGITAL2](#), [DIGITAL3](#), [DIGITAL4](#), [DIGITAL5](#),  
    [DIGITAL6](#) }  
*These are the event codes for the Rail Driver's report message.*

### 8.24.1 Detailed Description

Namespace that holds the Raildriver Client class code.

Author

Robert Heller <heller@deepsoft.com>

### 8.24.2 Package provided

[RaildriverClient](#) 1.0.0

### 8.24.3 Typedef Documentation

#### 8.24.3.1 eventlist

```
list< raildriver::RaildriverEvents > raildriver::eventlist
```

List of event codes.

### 8.24.4 Enumeration Type Documentation

#### 8.24.4.1 RaildriverEvents

```
enum raildriver::RaildriverEvents
```

These are the event codes for the Rail Driver's report message.

There is a code for each of the thirteen bytes in the report buffer.

Enumerator

REVERSER	Reverser lever. This is a value between 0 and 255 representing the position of the reverser lever.
THROTTLE	Throttle lever. This is a value between 0 and 255 representing the position of the throttle / dynamic brake lever.
AUTOBRAKE	Automatic Brake lever. This is a value between 0 and 255 representing the position of the automatic brake lever.
INDEPENDBRK	Independent Brake lever. This is a value between 0 and 255 representing the position of the independent brake lever.

## Enumerator

BAILOFF	Independent Brake bail off. This is a value between 0 and 255 representing the position of the independent brake lever bail off.
WIPER	Wiper switch. This is a value between 0 and 255 representing the position of the wiper switch.
HEADLIGHT	Headlight switch. This is a value between 0 and 255 representing the position of the headlight switch.
DIGITAL1	Blue Buttons 1-8. This is a bitfield representing 8 of the generic "blue" buttons.
DIGITAL2	Blue Buttons 9-16. This is a bitfield representing 8 of the generic "blue" buttons.
DIGITAL3	Blue Buttons 17-24. This is a bitfield representing 8 of the generic "blue" buttons.
DIGITAL4	Blue Buttons 25-28, Zoom, Pan. This is a bitfield representing the last 4 of the generic "blue" buttons, the zoom rocker, and one-half of the pan (2d) rocker.
DIGITAL5	Pan, Cab Buttons. This is a bitfield representing the second half of the pan (2d) rocker, and several of the two of the cab rocker switches.
DIGITAL6	Cab Buttons, Whistle. This is a bitfield representing the cab buttons and the whistle lever.

## 8.25 ReadConfiguration Namespace Reference

The Read Configuration File code is contained in this namespace.

### Functions

- [ReadConfiguration](#) (filename, configurationArrayName)  
*This procedure reads in the configuration file named by the filename into the array named by configurationArrayName.*
- [IsEven](#) (i)  
*Checks if its argument is an even number.*
- [WriteConfiguration](#) (filename, configurationArrayName)  
*This procedure writes the configuration contained in configurationArrayName to the file named by the filename.*
- [ConfigurationType](#) (...)  
*This macro defines the body of a snit::type that implements a program's global configuration (or preferences).*

### 8.25.1 Detailed Description

The Read Configuration File code is contained in this namespace.

#### Author

Robert Heller <heller@deepsoft.com>

### 8.25.2 Package provided

[ReadConfiguration](#) 1.0

### 8.25.3 Function Documentation

#### 8.25.3.1 ConfigurationType()

```
ReadConfiguration::ConfigurationType (
    ... )
```

This macro defines the body of a `snit::type` that implements a program's global configuration (or preferences).

The argument list is a set of configuration variable definition lists. Each list contains four elements: the label, the key list name (a one or two element list), the type (one of `directory`, `infile`, `outfile`, `string`, `enumerated`, `integer`, `double`, or `color`), and the default value. Enumerated types have an additional (fifth) element, the set of possible values. Numerical types (`double` and `integer`) have a range of values as a fifth element. This macro should only be called inside a `snit::type` definition.

The configuration (aka preferences) are stored in the user's home directory. The file name under UNIX (including Mac↔OSX) starts with a dot and contains the application rootname (from `argv0`). Under MS-Windows, the file name does not start with a dot. Instead `.rc` is appended.

#### Parameters

...	The configuration variable definitions.
-----	---

Type methods defined:

- `load` Load the configuration.
- `save` Save the configuration.
- `edit` Edit the configuration with a popup dialog.
- `getkeyoption` Get a keyed option. Takes two arguments.
- `getoption` Get a non-keyed option. Takes one argument.
- `getanonymous` Gets the anonymous option. Takes no arguments.

#### Author

Robert Heller <heller@deepsoft.com>

#### 8.25.3.2 IsEven()

```
ReadConfiguration::IsEven (
    i )
```

Checks if its argument is an even number.

**Parameters**

<i>i</i>	Value to check.
----------	-----------------

**Author**

Robert Heller <heller@deepsoft.com>

**8.25.3.3 ReadConfiguration()**

```
ReadConfiguration::ReadConfiguration (
    filename ,
    configurationArrayName )
```

This procedure reads in the configuration file named by the filename into the array named by configurationArrayName.

**Parameters**

<i>filename</i>	The name of the configuration file.
<i>configurationArrayName</i>	The name of the array to hold the configuration.

**Author**

Robert Heller <heller@deepsoft.com>

**8.25.3.4 WriteConfiguration()**

```
ReadConfiguration::WriteConfiguration (
    filename ,
    configurationArrayName )
```

This procedure writes the configuration contained in configurationArrayName to the file named by the filename.

**Parameters**

<i>filename</i>	The name of the configuration file.
<i>configurationArrayName</i>	The name of the array holding the configuration.

## Author

Robert Heller <heller@deepsoft.com>

## 8.26 TTSupport Namespace Reference

Time Table Support Namespace.

### Classes

- class [Cab](#)  
*This class maintains information about cabs.*
- struct [eqstr](#)
- struct [hash](#)  
*Option hash map, used for Print options.*
- class [Occupied](#)  
*This class records a train sitting on a storage track during a specified time frame.*
- class [PathName](#)  
*A Class that portably represents a pathname.*
- class [Station](#)  
*The [Station](#) class implements a station.*
- class [StationTimes](#)  
*[Station](#) times class, used by the LaTeX generator methods.*
- class [Stop](#)  
*This class implements a stop.*
- class [StorageTrack](#)  
*The [StorageTrack](#) class implements a storage track.*
- class [TimeRange](#)  
*The [TimeRange](#) class implements a range of times.*
- class [TimeTableSystem](#)  
*This is the main Time Table Class.*
- class [Train](#)  
*This class implements a train.*

### Typedefs

- typedef vector< double > [doubleVector](#)  
*A Vector of doubles.*
- typedef std::unordered\_map< const char \*, std::string, [hash](#), [eqstr](#) > [OptionHashMap](#)
- typedef list< [Train](#) \* > [TrainList](#)  
*List of trains.*
- typedef map< string, [StationTimes](#), less< string > > [TrainStationTimes](#)  
*Map of station times, indexed by train number.*
- typedef map< int, [TrainStationTimes](#), less< int > > [TrainTimesAtStation](#)  
*Map of maps of station times, indexed by station index.*

- typedef list< string > [StringList](#)  
*List of strings.*
- typedef map< [TimeRange](#), [Occupied](#), less< [TimeRange](#) > > [OccupiedMap](#)  
*The [Occupied](#) Map type, ordered by time ranges.*
- typedef map< string, [StorageTrack](#), less< string > > [StorageTrackMap](#)  
*Storage track map.*
- typedef vector< [Station](#) > [StationVector](#)  
*[Station](#) Vector.*
- typedef vector< [Stop](#) > [StopVector](#)  
*A vector of stops.*
- typedef map< string, [Train](#) \*, less< string > > [TrainNumberMap](#)  
*[Train](#) number map, indexed by train number (symbol).*
- typedef map< string, [Cab](#) \*, less< string > > [CabNameMap](#)  
*[Cab](#) name map, cabs indexed by name.*
- typedef vector< string > [stringVector](#)  
*A Vector of strings.*

## Functions

- const char \* [StringListToString](#) (const [StringList](#) &list)  
*Convert a list of strings to a flat string.*
- bool [StringListFromString](#) (string strlinList, [StringList](#) &result)  
*Convert a flat string to a list of strings.*

### 8.26.1 Detailed Description

Time Table Support Namespace.

Author

Robert Heller <heller@deepsoft.com>

### 8.26.2 Tcl Package Provided

Ttclasses 1.0.2

### 8.26.3 Library Provided

libttclasses 1.0.2

### 8.26.4 Typedef Documentation

#### 8.26.4.1 doubleVector

```
typedef vector<double> TTSupport::doubleVector
```

A Vector of doubles.

Used as a vector of layover times.

##### Author

Robert Heller <heller@deepsoft.com>

#### 8.26.4.2 OptionHashMap

```
typedef std::unordered_map<const char*, std::string, hash, eqstr> TTSupport::OptionHashMap
```

#### 8.26.4.3 StringList

```
typedef list<string> TTSupport::StringList
```

List of strings.

This is a simple linked list of strings, used in various places.

##### Author

Robert Heller <heller@deepsoft.com>

#### 8.26.4.4 stringVector

```
typedef vector<string> TTSupport::stringVector
```

A Vector of strings.

Used as the list of path list in a [PathName](#) instance.



#### 8.26.4.5 TrainList

```
typedef list<Train*> TTSupport::TrainList
```

List of trains.

Simple linked list of trains, used for passing train lists around.

##### Author

Robert Heller <heller@deepsoft.com>

#### 8.26.4.6 TrainStationTimes

```
typedef map<string, StationTimes, less<string> > TTSupport::TrainStationTimes
```

Map of station times, indexed by train number.

These are the individual rows of the time table. The train number (symbol) is the column index. Each of these rows is for a single station. This is a sparse vector, since not all trains stop at (or go past) all stations. The omitted elements result in blank cells in the output table.

##### Author

Robert Heller <heller@deepsoft.com>

#### 8.26.4.7 TrainTimesAtStation

```
typedef map<int, TrainStationTimes, less<int> > TTSupport::TrainTimesAtStation
```

Map of maps of station times, indexed by station index.

This is the whole time table. The station index is the row index. This is a sparse vector, since not all trains stop at (or go past) all stations. The omitted elements result in blank cells in the output table.

##### Author

Robert Heller <heller@deepsoft.com>

### 8.26.5 Function Documentation

#### 8.26.5.1 StringListFromString()

```
bool TTSupport::StringListFromString (
    string strlinList,
    StringList & result )
```

Convert a flat string to a list of strings.

Returns false if there was a syntax error.

**Parameters**

<i>strlinList</i>	The input string.
<i>result</i>	The output list.

**Author**

Robert Heller <heller@deepsoft.com>

**8.26.5.2 StringListToString()**

```
const char* TTSupport::StringListToString (
    const StringList & list )
```

Convert a list of strings to a flat string.

The result is comma separated and each string is enclosed in quote characters ("). If a string contains a quote character or a backslash, the character is quoted with a backslash.

**Parameters**

<i>list</i>	The list of strings.
-------------	----------------------

**Author**

Robert Heller <heller@deepsoft.com>

## 8.27 xpressnet Namespace Reference

Namespace that holds the [XPressNet](#) interface code.

**Classes**

- class [AccessoryDecoderInformation](#)  
*Accessory decoder information.*
- class [CommandStationResponse](#)  
*General response class.*
- class [CommandStationStatus](#)  
*Command station status.*
- class [DoubleHeaderInformation](#)  
*Double header information.*

- class [DoubleHeaderMuError](#)  
*Double header or MU error.*
- class [FunctionStatus](#)  
*Function status.*
- class [LI100Message](#)  
*LI100 messages.*
- class [LI100VersionNumbers](#)  
*LI100 Version Numbers.*
- class [LI101XPressNetAddress](#)  
*LI101 XPress Net Address.*
- class [LocomotiveAddress](#)  
*Locomotive address.*
- class [LocomotiveInformation](#)  
*Locomotive information.*
- class [ServiceModeResponse](#)  
*Service mode response.*
- class [SoftwareVersion](#)  
*Software version.*
- class [XPressNet](#)  
*Main XPressNet interface class.*
- class [XpressNetEvent](#)  
*XPressNet Event class.*

## Typedefs

- typedef int [nibble](#)  
*A 4 bit unsigned integer.*
- typedef int [ubyte](#)  
*An 8 bit unsigned integer.*
- typedef int [DecoderLongAddress](#)  
*Decoder address, an unsigned 14 bit integer.*
- typedef int [ElementAddress](#)  
*A 2 bit unsigned integer.*
- typedef int [S\\_14](#)  
*14 Speed steps.*
- typedef int [S\\_27](#)  
*27 Speed steps.*
- typedef int [S\\_28](#)  
*28 Speed steps.*
- typedef int [S\\_128](#)  
*128 Speed steps.*
- typedef int [u10](#)  
*An unsigned 10 bit integer.*
- typedef int [u3](#)  
*An unsigned 3 bit integer.*
- typedef int [u7](#)  
*An unsigned 7 bit integer.*
- typedef int [ConsistAddress](#)  
*Multi-unit Address.*

## Enumerations

- enum `TypeCode` {  
`NO_RESPONSE_AVAILABLE`, `NORMAL_OPERATION_RESUMED`, `TRACK_POWER_OFF`, `EMERGENCY_STOP`,  
`SERVICE_MODE_ENTRY`, `PROGRAMMING_INFO_SHORT_CIRCUIT`, `PROGRAMMING_INFO_DATA_BYTE_NOT_FOUND`,  
`PROGRAMMING_INFO_COMMAND_STATION_BUSY`, `PROGRAMMING_INFO_COMMAND_STATION_READY`, `SERVICE_MODE_RESPONSE`, `SOFTWARE_VERSION`, `COMMAND_STATION_STATUS`,  
`TRANSFER_ERRORS`, `COMMAND_STATION_BUSY`, `INSTRUCTION_NOT_SUPPORTED`, `ACCESSORY_DECODER_INFORMATION`,  
`LOCOMOTIVE_INFORMATION`, `FUNCTION_STATUS`, `LOCOMOTIVE_ADDRESS`, `DOUBLE_HEADER_INFORMATION`,  
`DOUBLE_HEADER_MU_ERROR`, `LI100_MESSAGE`, `LI100_VERSION`, `LI101_XPRESSNET_ADDRESS` }  
*Response types.*
- enum `PowerUpMode` { `Manual`, `Automatic` }  
*Power up modes.*
- enum `NibbleCode` { `Lower`, `Upper` }  
*Accessory nibble code.*
- enum `SpeedStepModeCode` { `S14`, `S27`, `S28`, `S128` }  
*Speed step mode code.*
- enum `DirectionCode` { `Forward`, `Reverse` }  
*Direction flag.*
- enum `ErrorTypeCode` {  
`NotOperatedOr0`, `UsedByAnotherDevice`, `UsedInANotherDHMU`, `SpeedNotZero`,  
`NotMU`, `NotMUBaseAddress`, `CantDelete`, `StackFull` }  
*Error type code.*
- enum `MessageTypeCode` {  
`ErrorBetweenLI100AndPC`, `ErrorBetweenLI100AndCommandStation`, `UnknownCommunicationsError`, `Success`,  
`NoTimeslot`, `BufferOverflow`, `Other` }  
*Message type code.*

### 8.27.1 Detailed Description

Namespace that holds the `XPressNet` interface code.

This is a cross-platform implementation the `XPressNet` serial port interface. Based on documentation provided by Lenz Elektronik GMBH (6/2003 third edition). This code works with Tcl 8.4 and later and uses SNIT to implement the classes as `snit::types`.

Basically, the way this code works is to use a class to interface to the serial port attached to one of Lenz's serial port adapters (LI100, LI100F, or LI101). This code should also work with the LiUSB interface as well.

#### Author

Robert Heller <heller@deepsoft.com>

### 8.27.2 Package provided

Xpressnet 2.0.0

### 8.27.3 Typedef Documentation

#### 8.27.3.1 ConsistAddress

int `xpressnet::ConsistAddress`

Multi-unit Address.

#### 8.27.3.2 DecoderLongAddress

short int `xpressnet::DecoderLongAddress`

Decoder address, an unsigned 14 bit integer.

#### 8.27.3.3 ElementAddress

int `xpressnet::ElementAddress`

A 2 bit unsigned integer.

#### 8.27.3.4 nibble

int `xpressnet::nibble`

A 4 bit unsigned integer.

#### 8.27.3.5 S\_128

int `xpressnet::S_128`

128 Speed steps.

**8.27.3.6 S\_14**

int `xpressnet::S_14`

14 Speed steps.

**8.27.3.7 S\_27**

int `xpressnet::S_27`

27 Speed steps.

**8.27.3.8 S\_28**

int `xpressnet::S_28`

28 Speed steps.

**8.27.3.9 u10**

int `xpressnet::u10`

An unsigned 10 bit integer.

**8.27.3.10 u3**

int `xpressnet::u3`

An unsigned 3 bit integer.

**8.27.3.11 u7**

int `xpressnet::u7`

An unsigned 7 bit integer.

## 8.27.3.12 ubyte

```
unsigned char xpressnet::ubyte
```

An 8 bit unsigned integer.

## 8.27.4 Enumeration Type Documentation

## 8.27.4.1 DirectionCode

```
enum xpressnet::DirectionCode
```

Direction flag.

Enumerator

Forward	Forward.
Reverse	Reverse.

## 8.27.4.2 ErrorTypeCode

```
enum xpressnet::ErrorTypeCode
```

Error type code.

Enumerator

NotOperatedOr0	One of the locomotives has not been operated by the XpressNet device assembling the Double Header/Multi Unit or locomotive 0 was selected.
UsedByAnotherDevice	One of the locomotives of the Double Header/Multi Unit is being used by another XpressNet device.
UsedInANotherDHMU	One of the locomotives is already in another Double Header/Multi Unit.
SpeedNotZero	The speed of one of the locomotives is not zero.
NotMU	The locomotive is not a multi-unit.
NotMUBaseAddress	The locomotive is not a multi-unit base address.
CantDelete	It is not possible to delete the locomotive.
StackFull	The command station stack is full.

### 8.27.4.3 MessageTypeCode

enum `xpressnet::MessageTypeCode`

Message type code.

#### Enumerator

ErrorBetweenLI100AndPC	Error occured between the interface and the PC. (Timeout durring data communication with the PC.)
ErrorBetweenLI100AndCommandStation	Error occured between the interface and the command station. (Timeout durring data communication with the command station.)
UnknownCommunicationsError	Unknown communication error. (Command station addressed the LI100 with request for acknowledgement.)
Success	Instruction was successfully sent to the command station or normal operations have resumed after a timeout.
NoTimeslot	The command station is no longer providing the LI100 a timeslot for communication.
BufferOverflow	Buffer overflow in the LI100.
Other	Other messages (undefined).

### 8.27.4.4 NibbleCode

enum `xpressnet::NibbleCode`

Accessory nibble code.

#### Enumerator

Lower	Lower nibble.
Upper	Upper nibble.

### 8.27.4.5 PowerUpMode

enum `xpressnet::PowerUpMode`

Power up modes.

#### Enumerator

Manual	Manual mode.
Automatic	Automatic mode.



## 8.27.4.6 SpeedStepModeCode

```
enum xpressnet::SpeedStepModeCode
```

Speed step mode code.

## Enumerator

S14	14 speed step mode.
S27	27 speed step mode.
S28	28 speed step mode.
S128	128 speed step mode.

## 8.27.4.7 TypeCode

```
enum xpressnet::TypeCode
```

Response types.

## Enumerator

NO_RESPONSE_AVAILABLE	No response available.
NORMAL_OPERATION_RESUMED	Normal operation resumed.
TRACK_POWER_OFF	Track power off.
EMERGENCY_STOP	Emergency stop.
SERVICE_MODE_ENTRY	Service mode entry.
PROGRAMMING_INFO_SHORT_CIRCUIT	Programming info. "short-circuit".
PROGRAMMING_INFO_DATA_BYTE_NOT_FOUND	Programming info. "data byte not found".
PROGRAMMING_INFO_COMMAND_STATION_BUSY	Programming info. "command station busy".
PROGRAMMING_INFO_COMMAND_STATION_READY	Programming info. "command station ready".
SERVICE_MODE_RESPONSE	Service mode response.
SOFTWARE_VERSION	Software version.
COMMAND_STATION_STATUS	Command station status.
TRANSFER_ERRORS	Transfer errors.
COMMAND_STATION_BUSY	Command station busy.
INSTRUCTION_NOT_SUPPORTED	Instruction not supported by command station.
ACCESSORY_DECODER_INFORMATION	Accessory decoder information.
LOCOMOTIVE_INFORMATION	Locomotive information.
FUNCTION_STATUS	Function status.
LOCOMOTIVE_ADDRESS	Locomotive address.
DOUBLE_HEADER_INFORMATION	Double header information.

## Enumerator

DOUBLE_HEADER_MU_ERROR	Double header or MU error.
LI100_MESSAGE	LI100 Messages.
LI100_VERSION	LI100 Version Numbers.
LI101_XPRESSNET_ADDRESS	LI101 <a href="#">XPressNet</a> Address.

## Chapter 9

# Class Documentation

### 9.1 xpressnet::AccessoryDecoderInformation Class Reference

Accessory decoder information.

#### Public Member Functions

- [AccessoryDecoderInformation](#) (name, count,...)  
*Constructor.*
- [NumberOfFeedbackElements](#) ()  
*Return the number of feedback elements.*
- [Address](#) (index)  
*Return address.*
- [Completed](#) (index)  
*Return completed flag.*
- [AccessoryType](#) (index)  
*Return accessory type.*
- [Nibble](#) (index)  
*Return nibble code.*
- [TurnoutStatus](#) (index, [nibble](#))  
*Return turnout status.*

#### Private Attributes

- [\\_numberOfFeedbackElements](#)  
*Number of Accessory Decoder feedback elements.*
- [\\_address](#)  
*Address value.*
- [\\_completed](#)  
*Completion flag.*

- [\\_accessory\\_type](#)  
*Accessory type.*
- [\\_nibble](#)  
*Nibble value.*
- [\\_t1](#)  
*Lower nibble turnout status.*
- [\\_t2](#)  
*Upper nibble turnout status.*

### 9.1.1 Detailed Description

Accessory decoder information.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.1.2 Constructor & Destructor Documentation

#### 9.1.2.1 AccessoryDecoderInformation()

```
xpressnet::AccessoryDecoderInformation::AccessoryDecoderInformation (
    name ,
    count ,
    ... )
```

Constructor.

#### Parameters

<i>count</i>	Number of Accessory Decoder feedback elements (1 through 7).
<i>args</i>	Address and data bytes.

### 9.1.3 Member Function Documentation

#### 9.1.3.1 AccessoryType()

```
xpressnet::AccessoryDecoderInformation::AccessoryType (
    index )
```

Return accessory type.

**Parameters**

<i>index</i>	Element index.
--------------	----------------

**9.1.3.2 Address()**

```
xpressnet::AccessoryDecoderInformation::Address (
    index )
```

Return address.

**Parameters**

<i>index</i>	Element index.
--------------	----------------

**9.1.3.3 Completed()**

```
xpressnet::AccessoryDecoderInformation::Completed (
    index )
```

Return completed flag.

**Parameters**

<i>index</i>	Element index.
--------------	----------------

**9.1.3.4 Nibble()**

```
xpressnet::AccessoryDecoderInformation::Nibble (
    index )
```

Return nibble code.

**Parameters**

<i>index</i>	Element index.
--------------	----------------

#### 9.1.3.5 NumberOfFeedbackElements()

```
xpressnet::AccessoryDecoderInformation::NumberOfFeedbackElements ( )
```

Return the number of feedback elements.

#### 9.1.3.6 TurnoutStatus()

```
xpressnet::AccessoryDecoderInformation::TurnoutStatus (
    index ,
    nibble )
```

Return turnout status.

##### Parameters

<i>index</i>	Element index.
<i>nibble</i>	Which turnout?

### 9.1.4 Member Data Documentation

#### 9.1.4.1 \_accessory\_type

```
xpressnet::AccessoryDecoderInformation::_accessory_type [private]
```

Accessory type.

#### 9.1.4.2 \_address

```
xpressnet::AccessoryDecoderInformation::_address [private]
```

Address value.

#### 9.1.4.3 `_completed`

`xpressnet::AccessoryDecoderInformation::_completed` [private]

Completion flag.

#### 9.1.4.4 `_nibble`

`xpressnet::AccessoryDecoderInformation::_nibble` [private]

Nibble value.

#### 9.1.4.5 `_numberOfFeedbackElements`

`xpressnet::AccessoryDecoderInformation::_numberOfFeedbackElements` [private]

Number of Accessory Decoder feedback elements.

#### 9.1.4.6 `_t1`

`xpressnet::AccessoryDecoderInformation::_t1` [private]

Lower nibble turnout status.

#### 9.1.4.7 `_t2`

`xpressnet::AccessoryDecoderInformation::_t2` [private]

Upper nibble turnout status.

## 9.2 `Instruments::AnalogClock` Class Reference

Analog clock instrument.

### Public Member Functions

- [`AnalogClock`](#) (name, \_canvas,...)  
*Constructor – initialize an analog clock.*
- [`~AnalogClock`](#) ()  
*Destructor – free up all resources.*
- [`settime`](#) (hour, minute)  
*Method to set the time.*

### 9.2.1 Detailed Description

Analog clock instrument.



## Parameters

<code>_canvas</code>	The canvas to draw the analog clock on.
<code>...</code>	Options: <ul style="list-style-type: none"> <li>• <code>-x</code> The X coordinate of the instrument (default 0).</li> <li>• <code>-y</code> The Y coordinate of the instrument (default 0).</li> <li>• <code>-size</code> The size of the instrument (default 100).</li> <li>• <code>-label</code> The label of the instrument (default Clock).</li> <li>• <code>-labelcolor</code> The color of the label (default black).</li> <li>• <code>-labelfont</code> The font of the label (default {Times 14 bold}).</li> <li>• <code>-background</code> The background color of the instrument (default blue).</li> <li>• <code>-outline</code> The outline color of the instrument (default black).</li> <li>• <code>-scaleback</code> The background color of the scale (default white).</li> <li>• <code>-scaleticks</code> The color of the scale ticks (default black).</li> <li>• <code>-fontfamily</code> The font family used on the instrument (default Courier).</li> <li>• <code>-hubcolor</code> The color of the hub (default black).</li> <li>• <code>-minutehandcolor</code> The color of the minute hand (default black).</li> <li>• <code>-hourhandcolor</code> The color of the hour hand (default black).</li> </ul>

## Author

Robert Heller <heller@deepsoft.com>

## 9.2.2 Constructor &amp; Destructor Documentation

## 9.2.2.1 AnalogClock()

```
Instruments::AnalogClock::AnalogClock (
    name ,
    _canvas ,
    ... )
```

Constructor – initialize an analog clock.

## Parameters

<code>_canvas</code>	The canvas to draw the <a href="#">DialInstrument</a> on.
<code>...</code>	Option list.

9.2.2.2 `~AnalogClock()`

```
Instruments::AnalogClock::~~AnalogClock ( )
```

Destructor – free up all resources.

## 9.2.3 Member Function Documentation

9.2.3.1 `settime()`

```
Instruments::AnalogClock::settime (
    hour ,
    minute )
```

Method to set the time.

## Parameters

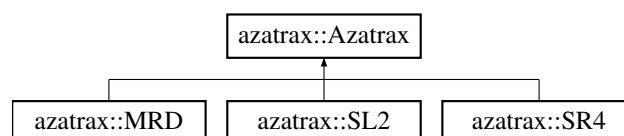
<i>hour</i>	The hour, an integer between 1 and 12.
<i>minute</i>	The minute, an integer between 0 and 59.

9.3 `azatrax::Azatrax` Class Reference

[Azatrax](#) I/O Class.

```
#include <Azatrax.h>
```

Inheritance diagram for `azatrax::Azatrax`:



## Classes

- struct [StateDataPacket](#)

*Raw USB Data Packet.*

## Public Types

- enum [commands](#) {  
[cmd\\_SetChan1](#) =0x31, [cmd\\_SetChan2](#) =0x32, [cmd\\_ClearExternallyChanged](#) =0x34, [cmd\\_DisableExternal](#) =0x37,  
[cmd\\_EnableExternal](#) =0x38, [cmd\\_Q1posQ2neg](#) =0x39, [cmd\\_Q1negQ2pos](#) =0x3A, [cmd\\_Q1Q2open](#) =0x3F,  
[cmd\\_Q3posQ4neg](#) =0x49, [cmd\\_Q3negQ4pos](#) =0x4A, [cmd\\_Q3Q4open](#) =0x4F, [cmd\\_RestoreLEDFunction](#) =0x50,  
[cmd\\_Identify\\_1](#) =0x51, [cmd\\_Identify\\_2](#) =0x52, [cmd\\_Identify\\_1\\_2](#) =0x53, [cmd\\_ResetStopwatch](#) =0x54,  
[cmd\\_GetStateData](#) =0x57, [cmd\\_OutputRelayBlink](#) =0x61, [cmd\\_OutputRelayOff](#) =0x67, [cmd\\_OutputRelayOn](#) =0x6B,  
[cmd\\_OutputRelayPulse](#) =0x6D, [cmd\\_OutputRelayInputControl](#) =0x6E }

*Command codes.*

- enum [DeviceConnectionList](#) { [\\_InitSize](#) = 10, [\\_GrowSize](#) = 10 }

*Device connection list constants.*

- enum { [idAzatraxVendor](#) = 0x265F, [idMRDProduct](#) = 0xfcb2, [idSL2Product](#) = 0xfca1, [idSR4Product](#) = 0xfca2 }

*Azatrax vendor and product codes.*

## Public Member Functions

- [~Azatrax](#) ()  
*Base destructor.*
- [ErrorCode RestoreLEDFunction](#) () const  
*Restore LED function.*
- [ErrorCode Identify\\_1](#) () const  
*Identify 1.*
- [ErrorCode GetStateData](#) ()  
*Get State Data.*
- [uint8\\_t PacketCount](#) () const  
*Packet count.*
- [const char \\* SerialNumber](#) () const  
*Serial Number.*
- [const char \\* MyProduct](#) () const  
*My product name.*
- [unsigned short int MyProductId](#) () const  
*My product code.*

## Static Public Member Functions

- static int [NumberOfOpenDevices](#) ()  
*Open device count.*
- static char \*\* [AllConnectedDevices](#) ()  
*List serial numbers of all connected devices.*
- static [Azatrax](#) \* [OpenDevice](#) (const char \*serialnumber, unsigned short int idProduct=0, char \*\*outmessage=NULL)  
*Open an [Azatrax](#) device by serial number.*
- static unsigned short int [ProductIdCode](#) (const char \*productName)  
*Product id code.*

## Protected Member Functions

- [Azatrax](#) (const char \*serialnumber, unsigned short int idProduct, char \*\*outmessage=NULL)  
*Base constructor.*
- [ErrorCode](#) [sendByte](#) (uint8\_t commandByte) const  
*Send a command byte.*
- [ErrorCode](#) [send2Bytes](#) (uint8\_t commandByte, uint8\_t byte2) const  
*Send a command byte with 1 data byte.*
- [ErrorCode](#) [send3Bytes](#) (uint8\_t commandByte, uint8\_t byte2, uint8\_t byte3) const  
*Send a command byte with 2 data bytes.*

## Static Protected Member Functions

- static bool [IsThisTheAzatraxWeAreLookingFor](#) (libusb\_device \*dev, const char \*serialnumber, unsigned short int idProduct)  
*Is this the device we want?*
- static unsigned short int [GetProductId](#) (libusb\_device \*dev, const char \*serialnumber, unsigned short int idProductMatch=0)  
*Return the matching product id for this device and serial number.*

## Protected Attributes

- struct libusb\_device\_handle \* [handle](#)  
*USB Device Handle.*
- char [mySerialNumber](#) [10]  
*Serial number buffer.*
- unsigned short int [myProductId](#)  
*Product id.*
- struct [azatrax::Azatrax::StateDataPacket](#) [stateDataPacket](#)

## Static Protected Attributes

- static int [deviceOpenCount](#)  
*Device open count.*

## Friends

- class [MRD](#)
- class [SL2](#)
- class [SR4](#)

### 9.3.1 Detailed Description

[Azatrax](#) I/O Class.

Base [Azatrax](#) interface class. From this base class, a class is derived that implements the device-specific functions. The base class only implements the shared core functions. It also includes a static method that implements a factor that creates device instances from device serial numbers.

There are two static methods, one that return the number of open devices (active instances) and another that returns a list of discovered serial numbers.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.3.2 Member Enumeration Documentation

#### 9.3.2.1 anonymous enum

anonymous enum

[Azatrax](#) vendor and product codes.

This is the vendor and product codes ([Azatrax](#)) was granted for their USB products.

#### Enumerator

idAzatraxVendor	<a href="#">Azatrax</a> vendor id.
idMRDProduct	MRD2 Product id.
idSL2Product	<a href="#">SL2</a> Product id.
idSR4Product	<a href="#">SR4</a> Product id.

#### 9.3.2.2 commands

enum [azatrax::Azatrax::commands](#)

Command codes.

## Enumerator

cmd_SetChan1	Set channel 1 (MRD2 only)
cmd_SetChan2	Set channel 2 (MRD2 only)
cmd_ClearExternallyChanged	Clear Externally Changed (MRD2 only)
cmd_DisableExternal	Disable External Changes (MRD2 only)
cmd_EnableExternal	Enable External Changes (MRD2 only)
cmd_Q1posQ2neg	(SL2 only) Sets output terminal Q1 to positive, Q2 to negative.
cmd_Q1negQ2pos	(SL2 only) Sets output terminal Q1 to negative, Q2 to positive.
cmd_Q1Q2open	(SL2 only) Outputs Q1 & Q2 both set to open circuit (disconnects switch machine #1)
cmd_Q3posQ4neg	(SL2 only) Sets output terminal Q3 to positive, Q4 to negative.
cmd_Q3negQ4pos	(SL2 only) Sets output terminal Q3 to negative, Q4 to positive.
cmd_Q3Q4open	(SL2 only) Outputs Q3 & Q4 both set to open circuit (disconnects switch machine #1)
cmd_RestoreLEDFunction	Restore LED Function (All devices)
cmd_Identify_1	Identify LED 1 (All devices)
cmd_Identify_2	Identify LED 2 (MRD2 only)
cmd_Identify_1_2	Identify both LEDs (MRD2 only)
cmd_ResetStopwatch	Reset Stopwatch (MRD2 only)
cmd_GetStateData	Get State Data (All devices)
cmd_OutputRelayBlink	3 bytes - Sets output relay contacts to blinking state. 2nd Byte is a bit map of the outputs to be affected: bits 7-4: ignored bit 3: 1 if Q4 affected bit 2: 1 if Q3 affected bit 1: 1 if Q2 affected bit 0: 1 if Q1 affected 3rd Byte is the output on/off rate (50% duty cycle): bits<7:2>: ignored bits<1:0>: 1,1 - one cycle / 2 seconds 1,0 - one cycle / second 0,1 - two cycles / second 0,0 - four cycles / second
cmd_OutputRelayOff	2 bytes - Sets output relay contacts to 'off' state. 2nd Byte is a bit map of the outputs to be affected: bits 7-4: ignored bit 3: 1 if Q4 affected bit 2: 1 if Q3 affected bit 1: 1 if Q2 affected bit 0: 1 if Q1 affected
cmd_OutputRelayOn	(SR4 only) 2 bytes - Sets output relay contacts to 'on' state. 2nd Byte is a bit map of the outputs to be affected: bits 7-4: ignored bit 3: 1 if Q4 affected bit 2: 1 if Q3 affected bit 1: 1 if Q2 affected bit 0: 1 if Q1 affected
cmd_OutputRelayPulse	(SR4 only) 3 bytes - Pulses output relay contacts to momentary 'on' state. 2nd Byte is a bit map of the outputs to be affected: bits 7-4: ignored bit 3: 1 if Q4 affected bit 2: 1 if Q3 affected bit 1: 1 if Q2 affected bit 0: 1 if Q1 affected 3rd Byte is pulse duration in 0.5 second increments: 0x00: no pulse generated 0x01: 0.5 sec 0x02: 1.0 sec ... 0xFE 63.0 sec 0xFF 63.5 sec
cmd_OutputRelayInputControl	(SR4 and SL2 only) 2 bytes - Enables/disables discrete input lines from controlling outputs For SL2: When enabled, I1 & I2 affect Q1 & Q2 (switch 1), I3 & I4 affect Q3 & Q4 (switch 2). 2nd Byte is a bit map of the outputs to be affected: bits 7-4: ignored bit 3: 1 enables I4, 0 disables bit 2: 1 enables I3, 0 disables bit 1: 1 enables I2, 0 disables bit 0: 1 enables I1, 0 disables For SR4: 2nd Byte is a bit map of the inputs: bits 7-4: ignored bit 3: 1 if I4 active causes Q4 on, else 0 (I4 does not affect Q4) bit 2: 1 if I3 active causes Q3 on, else 0 (I3 does not affect Q3) bit 1: 1 if I2 active causes Q2 on, else 0 (I2 does not affect Q2) bit 0: 1 if I1 active causes Q1 on, else 0 (I1 does not affect Q1)

### 9.3.2.3 DeviceConnectionList

```
enum azatrax::Azatrax::DeviceConnectionList
```

Device connection list constants.

Enumerator

<code>_InitSize</code>	
<code>_GrowSize</code>	

## 9.3.3 Constructor & Destructor Documentation

### 9.3.3.1 Azatrax()

```
azatrax::Azatrax::Azatrax (
    const char * serialnumber,
    unsigned short int idProduct,
    char ** outmessage = NULL ) [protected]
```

Base constructor.

Parameters

<i>serialnumber</i>	The serial number of the device to open.
<i>idProduct</i>	The product Id to look for.
<i>outmessage</i>	To hold an error message, if any.

### 9.3.3.2 ~Azatrax()

```
azatrax::Azatrax::~~Azatrax ( )
```

Base destructor.

## 9.3.4 Member Function Documentation



#### 9.3.4.1 AllConnectedDevices()

```
static char** azatrax::Azatrax::AllConnectedDevices ( ) [static]
```

List serial numbers of all connected devices.

Return a vector of serial number strings.

Referenced by NumberOfOpenDevices().

#### 9.3.4.2 GetProductId()

```
static unsigned short int azatrax::Azatrax::GetProductId (
    libusb_device * dev,
    const char * serialnumber,
    unsigned short int idProductMatch = 0 ) [static], [protected]
```

Return the matching product id for this device and serial number.

Checks if the dev is a [Azatrax](#) device, with the specified serial number, matching for product id.

##### Parameters

<i>dev</i>	libusb_device struct pointer (from libusb_get_device_list).
<i>serialnumber</i>	the serial number we are looking for.
<i>idProductMatch</i>	either an exact product id or 0 (means any product id)

##### Returns

The actual product id (device matched) or 0 (no match).

Referenced by ProductIdCode().

#### 9.3.4.3 GetStateData()

```
ErrorCode azatrax::Azatrax::GetStateData ( )
```

Get State Data.

Get State Data - retrieve 16 byte data packet from sensor to in memory copy.

Referenced by Identify\_1().

#### 9.3.4.4 Identify\_1()

```
ErrorCode azatrax::Azatrax::Identify_1 ( ) const [inline]
```

Identify 1.

Identify 1 - Flashes sensor 1's LED.

References `cmd_Identify_1`, `ErrorCode`, `GetStateData()`, and `sendByte()`.

#### 9.3.4.5 IsThisTheAzatraxWeAreLookingFor()

```
static bool azatrax::Azatrax::IsThisTheAzatraxWeAreLookingFor (
    libusb_device * dev,
    const char * serialnumber,
    unsigned short int idProduct ) [static], [protected]
```

Is this the device we want?

Check to see if this device is a [Azatrax](#) device and if it has the serial number we want.

##### Parameters

<i>dev</i>	libusb_device struct pointer (from <code>libusb_get_device_list</code> ).
<i>serialnumber</i>	the serial number we are looking for.
<i>idProduct</i>	the product id code.

Referenced by `ProductIdCode()`.

#### 9.3.4.6 MyProduct()

```
const char* azatrax::Azatrax::MyProduct ( ) const [inline]
```

My product name.

Return our Product name.

References `idMRDProduct`, `idSL2Product`, `idSR4Product`, and `myProductId`.

## 9.3.4.7 MyProductId()

```
unsigned short int azatrax::Azatrax::MyProductId ( ) const [inline]
```

My product code.

Return our Product Id.

References myProductId.

## 9.3.4.8 NumberOfOpenDevices()

```
static int azatrax::Azatrax::NumberOfOpenDevices ( ) [inline], [static]
```

Open device count.

Return the number of open devices.

References AllConnectedDevices(), deviceOpenCount, and OpenDevice().

## 9.3.4.9 OpenDevice()

```
static Azatrax* azatrax::Azatrax::OpenDevice (
    const char * serialnumber,
    unsigned short int idProduct = 0,
    char ** outmessage = NULL ) [static]
```

Open an [Azatrax](#) device by serial number.

## Parameters

<i>serialnumber</i>	the serial number to open.
<i>idProduct</i>	The product id code identifying the type of device to open. Passing 0 means open any <a href="#">Azatrax</a> device.
<i>outmessage</i>	To hold an error message, if any.

Referenced by NumberOfOpenDevices().

## 9.3.4.10 PacketCount()

```
uint8_t azatrax::Azatrax::PacketCount ( ) const [inline]
```

Packet count.

References `azatrax::Azatrax::StateDataPacket::packetCount`, and `stateDataPacket`.

#### 9.3.4.11 `ProductIdCode()`

```
static unsigned short int azatrax::Azatrax::ProductIdCode (
    const char * productName ) [inline], [static]
```

Product id code.

Return a product ID code.

##### Parameters

<i>productName</i>	The name of the product.
--------------------	--------------------------

References `ErrorCode`, `GetProductId()`, `idMRDProduct`, `idSL2Product`, `idSR4Product`, `IsThisTheAzatraxWeAreLookingFor()`, `send2Bytes()`, `send3Bytes()`, and `sendByte()`.

#### 9.3.4.12 `RestoreLEDFunction()`

```
ErrorCode azatrax::Azatrax::RestoreLEDFunction ( ) const [inline]
```

Restore LED function.

Restore LED function - On-board LEDs return to their normal function of indicating status of sensors 1 and 2.

References `cmd_RestoreLEDFunction`, and `sendByte()`.

#### 9.3.4.13 `send2Bytes()`

```
ErrorCode azatrax::Azatrax::send2Bytes (
    uint8_t commandByte,
    uint8_t byte2 ) const [protected]
```

Send a command byte with 1 data byte.

Sends one command byte on endpoint 01, with one data byte.

## Parameters

<i>commandByte</i>	The command byte to send.
<i>byte2</i>	The data byte to send.

Referenced by azatrax::SL2::OutputRelayInputControl(), azatrax::SR4::OutputRelayInputControl(), ProductIdCode(), azatrax::SR4::RelaysOff(), and azatrax::SR4::RelaysOn().

## 9.3.4.14 send3Bytes()

```
ErrorCode azatrax::Azatrax::send3Bytes (
    uint8_t commandByte,
    uint8_t byte2,
    uint8_t byte3 ) const [protected]
```

Send a command byte with 2 data bytes.

Sends one command byte on endpoint 01, with two data bytes.

## Parameters

<i>commandByte</i>	The command byte to send.
<i>byte2</i>	The first data byte to send.
<i>byte3</i>	The second data byte to send.

Referenced by azatrax::SR4::BlinkRelays(), ProductIdCode(), and azatrax::SR4::PulseRelays().

## 9.3.4.15 sendByte()

```
ErrorCode azatrax::Azatrax::sendByte (
    uint8_t commandByte ) const [protected]
```

Send a command byte.

Sends one command byte on endpoint 01.

## Parameters

<i>commandByte</i>	the command byte to send.
--------------------	---------------------------

Referenced by azatrax::MRD::ClearExternallyChanged(), azatrax::MRD::DisableExternal(), azatrax::MRD::EnableExternal(), Identify\_1(), azatrax::MRD::Identify\_1\_2(), azatrax::MRD::Identify\_2(), ProductIdCode(), azatrax::MRD::

::ResetStopwatch(), RestoreLEDFunction(), azatrax::MRD::SetChan1(), azatrax::MRD::SetChan2(), azatrax::SL2::SetQ1negQ2pos(), azatrax::SL2::SetQ1posQ2neg(), azatrax::SL2::SetQ1Q2open(), azatrax::SL2::SetQ3negQ4pos(), azatrax::SL2::SetQ3posQ4neg(), and azatrax::SL2::SetQ3Q4open().

#### 9.3.4.16 SerialNumber()

```
const char* azatrax::Azatrax::SerialNumber ( ) const [inline]
```

Serial Number.

Return our serial number.

References mySerialNumber.

### 9.3.5 Friends And Related Function Documentation

#### 9.3.5.1 MRD

```
friend class MRD [friend]
```

#### 9.3.5.2 SL2

```
friend class SL2 [friend]
```

#### 9.3.5.3 SR4

```
friend class SR4 [friend]
```

### 9.3.6 Member Data Documentation

#### 9.3.6.1 deviceOpenCount

```
int azatrax::Azatrax::deviceOpenCount [static], [protected]
```

Device open count.

This is used to decide if and when to call libusb\_init and libusb\_exit.

Referenced by NumberOfOpenDevices().

#### 9.3.6.2 handle

```
struct libusb_device_handle* azatrax::Azatrax::handle [protected]
```

USB Device Handle.

This is the USB device handle for the device.

#### 9.3.6.3 myProductId

```
unsigned short int azatrax::Azatrax::myProductId [protected]
```

Product id.

Referenced by MyProduct(), and MyProductId().

#### 9.3.6.4 mySerialNumber

```
char azatrax::Azatrax::mySerialNumber[10] [protected]
```

Serial number buffer.

This holds the serial number of the device.

Referenced by SerialNumber().

### 9.3.6.5 stateDataPacket

```
struct azatrax::Azatrax::StateDataPacket azatrax::Azatrax::stateDataPacket [protected]
```

Referenced by azatrax::MRD::AllowingExternalChanges(), azatrax::MRD::ExternallyChanged(), azatrax::MRD::HasRelays(), azatrax::SL2::Input\_1\_Enabled(), azatrax::SR4::Input\_1\_Enabled(), azatrax::SL2::Input\_2\_Enabled(), azatrax::SR4::Input\_2\_Enabled(), azatrax::SL2::Input\_3\_Enabled(), azatrax::SR4::Input\_3\_Enabled(), azatrax::SL2::Input\_4\_Enabled(), azatrax::SR4::Input\_4\_Enabled(), azatrax::MRD::Latch\_1(), azatrax::MRD::Latch\_2(), azatrax::SL2::Motor\_1\_Direction(), azatrax::SL2::Motor\_1\_State(), azatrax::SL2::Motor\_2\_Direction(), azatrax::SL2::Motor\_2\_State(), azatrax::MRD::OperatingMode(), PacketCount(), azatrax::SR4::Q1\_State(), azatrax::SR4::Q2\_State(), azatrax::SR4::Q3\_State(), azatrax::SR4::Q4\_State(), azatrax::MRD::ResetStatus(), azatrax::SL2::Sense\_1(), azatrax::MRD::Sense\_1(), azatrax::SR4::Sense\_1\_Latch(), azatrax::SR4::Sense\_1\_Live(), azatrax::SL2::Sense\_2(), azatrax::MRD::Sense\_2(), azatrax::SR4::Sense\_2\_Latch(), azatrax::SR4::Sense\_2\_Live(), azatrax::SL2::Sense\_3(), azatrax::SR4::Sense\_3\_Latch(), azatrax::SR4::Sense\_3\_Live(), azatrax::SL2::Sense\_4(), azatrax::SR4::Sense\_4\_Latch(), azatrax::SR4::Sense\_4\_Live(), azatrax::MRD::Stopwatch(), and azatrax::MRD::StopwatchTicking().

## 9.4 Parsers::BezierBody Class Reference

List of Bezier body lines (T, E, S, and C lines).

```
#include <BezierBody.h>
```

### Public Member Functions

- [BezierBody](#) ([BezierBodyElt](#) \*e, [BezierBody](#) \*n)  
*Basic constructor.*
- [TrackBody](#) \* [BezierEnds](#) ()  
*Create a track endpoint list.*
- int [BezierSegmentCount](#) ()  
*Count segments (S, C, and J lines).*
- const [BezierBodyElt](#) \* [Element](#) () const  
*Return current element.*

### Static Public Member Functions

- static [BezierBody](#) \* [ConsBezierBody](#) ([BezierBodyElt](#) \*trbe, [BezierBody](#) \*trb)  
*Alternative constructor function.*
- static void [CleanUpBezierBody](#) ([BezierBody](#) \*trb)  
*Free up memory.*

### Private Member Functions

- void [CleanUpElement](#) ()  
*Free up memory.*



## Private Attributes

- [BezierBodyElt](#) \* *element*  
*Current element.*
- [BezierBody](#) \* *next*  
*Next element.*

## Friends

- class [BezierBodyElt](#)
- class [TrackGraph](#)

### 9.4.1 Detailed Description

List of Bezier body lines (T, E, S, and C lines).

#### Author

Robert Heller <heller@deepsoft.com>

### 9.4.2 Constructor & Destructor Documentation

#### 9.4.2.1 BezierBody()

```
Parsers::BezierBody::BezierBody (  
    BezierBodyElt * e,  
    BezierBody * n ) [inline]
```

Basic constructor.

### 9.4.3 Member Function Documentation

#### 9.4.3.1 BezierEnds()

```
TrackBody* Parsers::BezierBody::BezierEnds ( ) [inline]
```

Create a track endpoint list.

References [element](#), [next](#), [Parsers::BezierBodyElt::theEnd](#), and [Parsers::BezierBodyElt::theType](#).

#### 9.4.3.2 BezierSegmentCount()

```
int Parsers::BezierBody::BezierSegmentCount ( ) [inline]
```

Count segments (S, C, and J lines).

References element, next, and Parsers::BezierBodyElt::theType.

#### 9.4.3.3 CleanUpBezierBody()

```
static void Parsers::BezierBody::CleanUpBezierBody (
    BezierBody * trb ) [inline], [static]
```

Free up memory.

References CleanUpElement(), element, and next.

#### 9.4.3.4 CleanUpElement()

```
void Parsers::BezierBody::CleanUpElement ( ) [inline], [private]
```

Free up memory.

References Parsers::BezierBodyElt::theEnd, and Parsers::BezierBodyElt::theType.

Referenced by CleanUpBezierBody().

#### 9.4.3.5 ConsBezierBody()

```
static BezierBody* Parsers::BezierBody::ConsBezierBody (
    BezierBodyElt * trbe,
    BezierBody * trb ) [inline], [static]
```

Alternative constructor function.

#### 9.4.3.6 Element()

```
const BezierBodyElt* Parsers::BezierBody::Element ( ) const [inline]
```

Return current element.

#### 9.4.4 Friends And Related Function Documentation

##### 9.4.4.1 BezierBodyElt

```
friend class BezierBodyElt [friend]
```

##### 9.4.4.2 TrackGraph

```
friend class TrackGraph [friend]
```

#### 9.4.5 Member Data Documentation

##### 9.4.5.1 element

```
BezierBodyElt* Parsers::BezierBody::element [private]
```

Current element.

Referenced by BezierEnds(), BezierSegmentCount(), and CleanUpBezierBody().

##### 9.4.5.2 next

```
BezierBody* Parsers::BezierBody::next [private]
```

Next element.

Referenced by BezierEnds(), BezierSegmentCount(), and CleanUpBezierBody().

### 9.5 Parsers::BezierBodyElt Class Reference

Bezier Body elements: T, E, S, and C lines are collected.

```
#include <BezierBody.h>
```

## Classes

- struct [Pos](#)  
*Position structure.*

## Public Types

- enum [BezierBodyEltType](#) { [None](#), [BezierEnd](#), [BezierStraightSegment](#), [BezierCurvedSegment](#) }  
*Element types.*

## Public Member Functions

- [BezierBodyElt](#) ()  
*Constructor.*
- [~BezierBodyElt](#) ()  
*Destructor.*
- [BezierBodyEltType TheType](#) () const  
*Type accessor.*
- int [GetStraightSegment](#) (float &x1, float &y1, float &x2, float &y2) const  
*Fetch straight segment data.*
- int [GetCurveSegment](#) (float &r, float &x, float &y, float &a0, float &a1) const  
*Fetch curve segment data.*

## Static Public Member Functions

- static void [InitTSegId](#) ()  
*Segment count initializer.*
- static [BezierBodyElt](#) \* [MakeTrackEnd](#) ([TrackBodyElt](#) \*tbe)  
*Create an endpoint (T or E lines).*
- static [BezierBodyElt](#) \* [MakeStraightSegment](#) (float x1, float y1, float x2, float y2)  
*Create a straight segment (S lines).*
- static [BezierBodyElt](#) \* [MakeCurveSegment](#) (float r, float x, float y, float a0, float a1)  
*Create a curve segment (C lines).*

## Private Attributes

- [BezierBodyEltType theType](#)  
*Element type.*
- [TrackBodyElt](#) \* [theEnd](#)  
*Pointer to T or E line data.*
- int [segmentId](#)  
*Segment index (S or C lines).*
- [Pos pos1](#)  
*First position.*

- [Pos pos2](#)  
*Second position.*
- float [radius](#)  
*A radius value.*
- float [ang0](#)  
*An angle value.*
- float [ang1](#)  
*Another angle value.*

### Static Private Attributes

- static int [segCount](#)  
*Counter for S and C segments.*

### Friends

- class [TrackGraph](#)
- class [BezierBody](#)

## 9.5.1 Detailed Description

Bezier Body elements: T, E, S, and C lines are collected.

others are discarded.

### Author

Robert Heller <heller@deepsoft.com>

## 9.5.2 Member Enumeration Documentation

### 9.5.2.1 BezierBodyEltType

enum [Parsers::BezierBodyElt::BezierBodyEltType](#)

Element types.

#### Enumerator

None	Placeholder.
BezierEnd	T or E line.
BezierStraightSegment	S line.
BezierCurvedSegment	C line.

### 9.5.3 Constructor & Destructor Documentation

#### 9.5.3.1 BezierBodyElt()

```
Parsers::BezierBodyElt::BezierBodyElt ( ) [inline]
```

Constructor.

#### 9.5.3.2 ~BezierBodyElt()

```
Parsers::BezierBodyElt::~~BezierBodyElt ( ) [inline]
```

Destructor.

### 9.5.4 Member Function Documentation

#### 9.5.4.1 GetCurveSegment()

```
int Parsers::BezierBodyElt::GetCurveSegment (
    float & r,
    float & x,
    float & y,
    float & a0,
    float & a1 ) const [inline]
```

Fetch curve segment data.

References Parsers::BezierBodyElt::Pos::x, and Parsers::BezierBodyElt::Pos::y.

#### 9.5.4.2 GetStraightSegment()

```
int Parsers::BezierBodyElt::GetStraightSegment (
    float & x1,
    float & y1,
    float & x2,
    float & y2 ) const [inline]
```

Fetch straight segment data.

References Parsers::BezierBodyElt::Pos::x, and Parsers::BezierBodyElt::Pos::y.

#### 9.5.4.3 InitTSegId()

```
static void Parsers::BezierBodyElt::InitTSegId ( ) [inline], [static]
```

Segment count initializer.

#### 9.5.4.4 MakeCurveSegment()

```
static BezierBodyElt* Parsers::BezierBodyElt::MakeCurveSegment (
    float r,
    float x,
    float y,
    float a0,
    float a1 ) [inline], [static]
```

Create a curve segment (C lines).

References ang0, ang1, pos1, radius, segmentId, theType, Parsers::BezierBodyElt::Pos::x, and Parsers::BezierBodyElt::Pos::y.

#### 9.5.4.5 MakeStraightSegment()

```
static BezierBodyElt* Parsers::BezierBodyElt::MakeStraightSegment (
    float x1,
    float y1,
    float x2,
    float y2 ) [inline], [static]
```

Create a straight segment (S lines).

References pos1, pos2, segmentId, theType, Parsers::BezierBodyElt::Pos::x, and Parsers::BezierBodyElt::Pos::y.

#### 9.5.4.6 MakeTrackEnd()

```
static BezierBodyElt* Parsers::BezierBodyElt::MakeTrackEnd (
    TrackBodyElt * tbe ) [inline], [static]
```

Create an endpoint (T or E lines).

References theEnd, and theType.

#### 9.5.4.7 TheType()

```
BezierBodyElementType Parsers::BezierBodyElt::TheType ( ) const [inline]
```

Type accessor.

### 9.5.5 Friends And Related Function Documentation

#### 9.5.5.1 BezierBody

```
friend class BezierBody [friend]
```

#### 9.5.5.2 TrackGraph

```
friend class TrackGraph [friend]
```

### 9.5.6 Member Data Documentation

#### 9.5.6.1 ang0

```
float Parsers::BezierBodyElt::ang0 [private]
```

An angle value.

Referenced by MakeCurveSegment().

#### 9.5.6.2 ang1

```
float Parsers::BezierBodyElt::ang1 [private]
```

Another angle value.

Referenced by MakeCurveSegment().



#### 9.5.6.3 pos1

```
Pos Parsers::BezierBodyElt::pos1 [private]
```

First position.

Referenced by MakeCurveSegment(), and MakeStraightSegment().

#### 9.5.6.4 pos2

```
Pos Parsers::BezierBodyElt::pos2 [private]
```

Second position.

Referenced by MakeStraightSegment().

#### 9.5.6.5 radius

```
float Parsers::BezierBodyElt::radius [private]
```

A radius value.

Referenced by MakeCurveSegment().

#### 9.5.6.6 segCount

```
int Parsers::BezierBodyElt::segCount [static], [private]
```

Counter for S and C segments.

#### 9.5.6.7 segmentId

```
int Parsers::BezierBodyElt::segmentId [private]
```

Segment index (S or C lines).

Referenced by MakeCurveSegment(), and MakeStraightSegment().

#### 9.5.6.8 theEnd

`TrackBodyElt*` `Parsers::BezierBodyElt::theEnd` [private]

Pointer to T or E line data.

Referenced by `Parsers::BezierBody::BezierEnds()`, `Parsers::BezierBody::CleanUpElement()`, and `MakeTrackEnd()`.

#### 9.5.6.9 theType

`BezierBodyEltType` `Parsers::BezierBodyElt::theType` [private]

Element type.

Referenced by `Parsers::BezierBody::BezierEnds()`, `Parsers::BezierBody::BezierSegmentCount()`, `Parsers::BezierBody::CleanUpElement()`, `MakeCurveSegment()`, `MakeStraightSegment()`, and `MakeTrackEnd()`.

## 9.6 TTSupport::Cab Class Reference

This class maintains information about cabs.

```
#include <Cab.h>
```

### Public Member Functions

- `Cab` (string name\_="", string color\_="")  
*Construct a new cab.*
- `~Cab` ()  
*Clean things up.*
- const char \* `Name` () const  
*Return the name of the cab.*
- const char \* `Color` () const  
*Return the color of the cab.*
- `Cab` (const `Cab` &other)  
*Copy constructor.*
- `Cab` & `operator=` (const `Cab` &other)  
*Assignment operator.*
- ostream & `Write` (ostream &stream) const  
*Write object to a stream.*
- istream & `Read` (istream &stream)  
*Read an object from a stream.*

## Private Attributes

- string `name`  
*The name of the cab.*
- string `color`  
*The color of the cab.*

### 9.6.1 Detailed Description

This class maintains information about cabs.

A cab has a color and a name.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.6.2 Constructor & Destructor Documentation

#### 9.6.2.1 Cab() [1/2]

```
TTSupport::Cab::Cab (  
    string name_ = "",  
    string color_ = "" ) [inline]
```

Construct a new cab.

#### Parameters

<code>name</code> ↔ —	The name of the new cab.
<code>color</code> ↔ —	The color of the cab.

References `color`, and `name`.

#### 9.6.2.2 ~Cab()

```
TTSupport::Cab::~~Cab ( ) [inline]
```

Clean things up.

### 9.6.2.3 Cab() [2/2]

```
TTSupport::Cab::Cab (
    const Cab & other ) [inline]
```

Copy constructor.

Create a new cab as a copy of an existing cab.

#### Parameters

<i>other</i>	The other cab.
--------------	----------------

References color, and name.

## 9.6.3 Member Function Documentation

### 9.6.3.1 Color()

```
const char* TTSupport::Cab::Color ( ) const [inline]
```

Return the color of the cab.

References color.

### 9.6.3.2 Name()

```
const char* TTSupport::Cab::Name ( ) const [inline]
```

Return the name of the cab.

References name.

### 9.6.3.3 operator=()

```
Cab& TTSupport::Cab::operator= (
    const Cab & other ) [inline]
```

Assignment operator.

Assign one cab to another cab.

**Parameters**

<i>other</i>	The other cab.
--------------	----------------

References `color`, `name`, `Read()`, `Icc::stream`, and `Write()`.

**9.6.3.4 Read()**

```
istream& TTSupport::Cab::Read (  
    istream & stream )
```

Read an object from a stream.

**Parameters**

<i>stream</i>	Stream to read from.
---------------	----------------------

Referenced by `operator=()`.

**9.6.3.5 Write()**

```
ostream& TTSupport::Cab::Write (  
    ostream & stream ) const
```

Write object to a stream.

**Parameters**

<i>stream</i>	Stream to write to.
---------------	---------------------

Referenced by `operator=()`.

**9.6.4 Member Data Documentation****9.6.4.1 color**

```
string TTSupport::Cab::color [private]
```

The color of the cab.

Referenced by Cab(), Color(), and operator=().

#### 9.6.4.2 name

```
string TTSupport::Cab::name [private]
```

The name of the cab.

Referenced by Cab(), Name(), and operator=().

## 9.7 Instruments::CabSignalLamp Class Reference

Cab signal lamp type.

### Public Member Functions

- [CabSignalLamp](#) (name, \_canvas,...)  
*Constructor – initialize a Cab Signal Lamp.*
- [~CabSignalLamp](#) ()  
*Destructor – free up all resources.*

### Private Member Functions

- [\\_ConfigureXY](#) (option, value)  
*Method to configure X or Y.*
- [\\_ConfigureSize](#) (option, value)  
*Method to configure size.*
- [\\_ConfigureFillColor](#) (option, value)  
*Method to configure a fill color.*
- [\\_ConfigureOutlineColor](#) (option, value)  
*Method to configure an outline color.*

### Private Attributes

- [canvas](#)  
*Canvas the lamp is on.*
- [sx](#)  
*X size offset.*
- [sy](#)  
*Y size offset.*

#### 9.7.1 Detailed Description

Cab signal lamp type.

## Parameters

<code>_canvas</code>	The canvas to draw the cab signal lamp on.
<code>...</code>	Options: <ul style="list-style-type: none"> <li>• <code>-x</code> The X coordinate of the instrument (default 0).</li> <li>• <code>-y</code> The Y coordinate of the instrument (default 0).</li> <li>• <code>-size</code> The size of the instrument (default 100).</li> <li>• <code>-color</code> The color of the lamp (default black).</li> <li>• <code>-outline</code> The outline color of the instrument (default black).</li> </ul>

## Author

Robert Heller <heller@deepsoft.com>

## 9.7.2 Constructor &amp; Destructor Documentation

## 9.7.2.1 CabSignalLamp()

```
Instruments::CabSignalLamp::CabSignalLamp (
    name ,
    _canvas ,
    ... )
```

Constructor – initialize a Cab Signal Lamp.

## Parameters

<code>_canvas</code>	The canvas to draw the <a href="#">DialInstrument</a> on.
<code>...</code>	Option list.

## 9.7.2.2 ~CabSignalLamp()

```
Instruments::CabSignalLamp::~~CabSignalLamp ( )
```

Destructor – free up all resources.

### 9.7.3 Member Function Documentation

#### 9.7.3.1 \_ConfigureFillColor()

```
Instruments::CabSignalLamp::_ConfigureFillColor (  
    option ,  
    value ) [private]
```

Method to configure a fill color.

##### Parameters

<i>option</i>	The name of the option to configure.
<i>value</i>	The new value.

#### 9.7.3.2 \_ConfigureOutlineColor()

```
Instruments::CabSignalLamp::_ConfigureOutlineColor (  
    option ,  
    value ) [private]
```

Method to configure an outline color.

##### Parameters

<i>option</i>	The name of the option to configure.
<i>value</i>	The new value.

#### 9.7.3.3 \_ConfigureSize()

```
Instruments::CabSignalLamp::_ConfigureSize (  
    option ,  
    value ) [private]
```

Method to configure size.

##### Parameters

<i>option</i>	The name of the option to configure.
<i>value</i>	The new value.



#### 9.7.3.4 \_ConfigureXY()

```
Instruments::CabSignalLamp::_ConfigureXY (
    option ,
    value ) [private]
```

Method to configure X or Y.

##### Parameters

<i>option</i>	The name of the option to configure.
<i>value</i>	The new value.

### 9.7.4 Member Data Documentation

#### 9.7.4.1 canvas

```
Instruments::CabSignalLamp::canvas [private]
```

Canvas the lamp is on.

#### 9.7.4.2 sx

```
Instruments::CabSignalLamp::sx [private]
```

X size offset.

#### 9.7.4.3 sy

```
Instruments::CabSignalLamp::sy [private]
```

Y size offset.

## 9.8 Icc::CanAlias Class Reference

Implements a CAN Alias.

### Public Member Functions

- [getNextAlias](#) ()  
*Compute next alias.*
- [\\_peelnid](#) (value)  
*Peel the Node ID into bytes and initializing the 48 bit random number seed for alias generation.*
- [CanAlias](#) (name,...)  
*Construct a CAN Alias.*
- [getMyAlias](#) ()  
*Return the current alias value.*
- [getMyNIDList](#) ()  
*Return the NID list.*

### Static Public Member Functions

- static [validate](#) (object)  
*Validate the object as a [CanAlias](#) object.*

### Public Attributes

- [lfsr1](#)  
*Sequence value, upper 24 bits.*
- [lfsr2](#)  
*Sequence value, lower 24 bits.*
- [nidlist](#)  
*The Node ID as a list of 6 bytes.*
- [myalias](#)  
*My node alias.*

### 9.8.1 Detailed Description

Implements a CAN Alias.

Options:

- -nid The Node ID that the computer will assume in the format of `hh : hh : hh : hh : hh : hh` which is a 48 bit number expressed as 6 pairs of hexadecimal digits separated by colons (:).

## 9.8.2 Constructor & Destructor Documentation

### 9.8.2.1 CanAlias()

```
lcc::CanAlias::CanAlias (
    name ,
    ... )
```

Construct a CAN Alias.

#### Parameters

<i>name</i>	The NodeID
...	Options <ul style="list-style-type: none"><li>-nid The Node ID that the computer will assume in the format of <code>hh:hh:hh:hh:hh:hh</code> which is a 48 bit number expressed as 6 pairs of hexadecimal digits separated by colons (:).</li></ul>

## 9.8.3 Member Function Documentation

### 9.8.3.1 \_peelnid()

```
lcc::CanAlias::_peelnid (
    value )
```

Peel the Node ID into bytes and initializing the 48 bit random number seed for alias generation.

### 9.8.3.2 getMyAlias()

```
lcc::CanAlias::getMyAlias ( )
```

Return the current alias value.

#### Returns

The 12 bit node id alias.

### 9.8.3.3 getMyNIDList()

```
lcc::CanAlias::getMyNIDList ( )
```

Return the NID list.

#### Returns

The 6 byte list containing the NID.

### 9.8.3.4 getNextAlias()

```
lcc::CanAlias::getNextAlias ( )
```

Compute next alias.

### 9.8.3.5 validate()

```
static lcc::CanAlias::validate (
    object ) [static]
```

Validate the object as a [CanAlias](#) object.

#### Parameters

<i>object</i>	A possible <a href="#">CanAlias</a> object.
---------------	---

## 9.8.4 Member Data Documentation

### 9.8.4.1 lfsr1

```
lcc::CanAlias::lfsr1
```

Sequence value, upper 24 bits.

#### 9.8.4.2 lfsr2

```
lcc::CanAlias::lfsr2
```

Sequence value, lower 24 bits.

#### 9.8.4.3 myalias

```
lcc::CanAlias::myalias
```

My node alias.

#### 9.8.4.4 nidlist

```
lcc::CanAlias::nidlist
```

The Node ID as a list of 6 bytes.

## 9.9 lcc::CANGridConnect Class Reference

Base class to connect to a CAN bus using GridConnect formatted message over.

### Public Member Functions

- [CANGridConnect](#) (name,...)  
*Constructor: create a connection to a Grid Connect CAN bus.*
- [getAliasOfNID](#) (nid)  
*Fetch the alias of a NID.*
- [getNIDofAlias](#) (alias)  
*Get the NID of the alias.*
- [getAllNIDs](#) ()  
*Get all known NIDs.*
- [getAllAliases](#) ()  
*Get all known aliases.*
- [updateAliasMap](#) (nid, alias)  
*Update the alias map with the specified Node ID and Alias.*
- [populateAliasMap](#) ()  
*Send an AME.*
- [setMessageHandler](#) (handler)  
*Set the message handler.*
- [sendMessage](#) (...)  
*Send a message on the OpenLCB bus.*
- [sendOpenLCBMessage](#) (message)  
*Send a message on the OpenLCB bus.*
- [reserveAlias](#) (canalias)  
*Reserve an alias.*

## Private Member Functions

- [\\_sendDatagram](#) (message)  
*Send a datagram message.*
- [\\_reserveMyAlias](#) ()  
*Reserve my alias.*
- [\\_timedout](#) ()  
*Timeout method.*
- [\\_flags0](#) (srcid, r, doff)  
*Method to deal with possible multipart messages, with particular handling of multi-part Simple Node Info messages.*
- [\\_messageReader](#) (message)  
*Handling incoming messages.*
- [\\_sendmessage](#) (canmessage)  
*Send a low-level CAN bus message using the Grid Connect format.*

## Static Private Member Functions

- static [listeq](#) (a, b)  
*Compare two lists.*
- static [getBits](#) (top, bottom, [bytelist](#))  
*Get the selected bitfield.*
- static [countNUL](#) (list)  
*Count NUL bytes in a byte buffer.*
- static [listeq](#) (a, b)  
*Compare two lists.*

## Private Attributes

- [parent](#)  
*Parent instance.*
- [gcmessage](#)  
*[GridConnectMessage](#) component.*
- [gcreply](#)  
*[GridConnectReply](#) component.*
- [mtidetail](#)  
*[MTIDetail](#) component.*
- [mtiheader](#)  
*[MTIHeader](#) component.*
- [canheader](#)  
*[CANHeader](#) component.*
- [messagehandler](#)  
*Message handler.*
- [datagrambuffers](#)  
*Datagram buffers.*
- [messagebuffers](#)  
*General message buffers (for multi frame messages)*

- [simplenodeflags](#)  
*Simple node info flags.*
- [mycanalias](#)  
*My [CanAlias](#) component.*
- [aliasMap](#)  
*Alias to NID map.*
- [nidMap](#)  
*NID to alias map.*
- [\\_timeout](#)  
*Timeout flag.*
- [\\_timeoutFlag](#)  
*Timeout or error message received flag.*

### Static Private Attributes

- static [NIDPATTERN](#)  
*The regexp for breaking up the Node ID into bytes.*

#### 9.9.1 Detailed Description

Base class to connect to a CAN bus using GridConnect formatted message over.

Options:

- -parent The parent instance. Needs to implement read, write, and readevent methods. This is a readonly option only processed at instance creation.
- -nid The Node ID that the computer will assume in the format of `hh:hh:hh:hh:hh:hh` which is a 48 bit number expressed as 6 pairs of hexadecimal digits separated by colons (:).
- -promiscuousmode Promiscuous mode flag. If true all messages are handled, whether they are addressed to this node or not.

#### 9.9.2 Constructor & Destructor Documentation

##### 9.9.2.1 CANGridConnect()

```
lcc::CANGridConnect::CANGridConnect (
    name ,
    ... )
```

Constructor: create a connection to a Grid Connect CAN bus.

Connect to the CAN bus via a Grid Connect CAN bus.

## Parameters

<i>name</i>	The name of the instance.
...	<p>The options:</p> <ul style="list-style-type: none"> <li>• <code>-parent</code> The parent instance. Contains the low-level read and write methods.</li> <li>• <code>-nid</code> The Node ID that the computer will assume in the format of <code>hh:hh:hh:hh:hh:hh</code> which is a 48 bit number expressed as 6 pairs of hexadecimal digits separated by colons (:).</li> <li>• <code>-eventhandler</code> This is a script prefix that is run on incoming messages. The current message as a binary <a href="#">CanMessage</a> is appended.</li> <li>• <code>-promiscuousmode</code> Promiscuous mode flag. If true all messages are handled, whether they are addressed to this node or not.</li> </ul>

### 9.9.3 Member Function Documentation

#### 9.9.3.1 `_flags0()`

```
lcc::CANGridConnect::_flags0 (
    srcid ,
    r ,
    doff ) [private]
```

Method to deal with possible multipart messages, with partitular handling of multi-part Simple Node Info messages.

## Parameters

<i>srcid</i>	The source alias of the message.
--------------	----------------------------------

#### 9.9.3.2 `_messageReader()`

```
lcc::CANGridConnect::_messageReader (
    message ) [private]
```

Handling incoming messages.

Handle control (CAN) messages here. OpenLCB messages are assembled possibly from multiple CAN messages and then dispatched to the upper level message handler.



#### 9.9.3.3 \_reserveMyAlias()

```
lcc::CANGridConnect::_reserveMyAlias ( ) [private]
```

Reserve my alias.

##### Returns

A boolean value indicating a successfully reserved alias (true) or failure (false).

#### 9.9.3.4 \_sendDatagram()

```
lcc::CANGridConnect::_sendDatagram (
    message ) [private]
```

Send a datagram message.

A possibly multi-part datagram message is sent.

##### Parameters

<i>message</i>	The OpenLCB message to send.
----------------	------------------------------

#### 9.9.3.5 \_sendmessage()

```
lcc::CANGridConnect::_sendmessage (
    canmessage ) [private]
```

Send a low-level CAN bus message using the Grid Connect format.

##### Parameters

<i>canmessage</i>	The (binary) CANMessage to send.
-------------------	----------------------------------

#### 9.9.3.6 \_timeout()

```
lcc::CANGridConnect::_timeout ( ) [private]
```

Timeout method.

Called on timeout.

#### 9.9.3.7 countNUL()

```
static lcc::CANGridConnect::countNUL (
    list ) [static], [private]
```

Count NUL bytes in a byte buffer.

##### Parameters

<i>list</i>	The list of bytes to search.
-------------	------------------------------

##### Returns

The number of NUL (0) bytes in the list.

#### 9.9.3.8 getAliasOfNID()

```
lcc::CANGridConnect::getAliasOfNID (
    nid )
```

Fetch the alias of a NID.

##### Parameters

<i>nid</i>	A full NID of the form hh:hh:hh:hh:hh:hh
------------	--

##### Returns

The node's alias or the empty string if not known.

#### 9.9.3.9 getAllAliases()

```
lcc::CANGridConnect::getAllAliases ( )
```

Get all known aliases.

##### Returns

All known aliases.

#### 9.9.3.10 getAllNIDs()

```
lcc::CANGridConnect::getAllNIDs ( )
```

Get all known NIDs.

##### Returns

All known NIDS.

#### 9.9.3.11 getBits()

```
static lcc::CANGridConnect::getBits (
    top ,
    bottom ,
    bytelist ) [static], [private]
```

Get the selected bitfield.

Extract the bits from a list of 6 8-bit (byte) numbers representing a 48 bit number.

##### Parameters

<i>top</i>	Topmost (highest) bit number.
<i>bottom</i>	Bottommost (lowest) bit number.
<i>bytelist</i>	List of 6 bytes.

##### Returns

An integer value.

#### 9.9.3.12 getNIDofAlias()

```
lcc::CANGridConnect::getNIDofAlias (
    alias )
```

Get the NID of the alias.

##### Parameters

<i>alias</i>	The alias to look up.
--------------	-----------------------

**Returns**

The NID of the alias or the empty string if not known.

**9.9.3.13 listeq()** [1/2]

```
static lcc::CANGridConnect::listeq (  
    a ,  
    b ) [static], [private]
```

Compare two lists.

Compares two lists for equality.

**Parameters**

<i>a</i>	First list to compare.
<i>b</i>	Second list to compare.

**Returns**

A boolean value: true if the lists are the same, false if not.

**9.9.3.14 listeq()** [2/2]

```
static lcc::CANGridConnect::listeq (  
    a ,  
    b ) [static], [private]
```

Compare two lists.

Compares two lists for equality.

**Parameters**

<i>a</i>	First list to compare.
<i>b</i>	Second list to compare.

**Returns**

A boolean value: true if the lists are the same, false if not.

#### 9.9.3.15 populateAliasMap()

```
lcc::CANGridConnect::populateAliasMap ( )
```

Send an AME.

#### 9.9.3.16 reserveAlias()

```
lcc::CANGridConnect::reserveAlias (
    canalias )
```

Reserve an alias.

Sends out CID messages and eventually RID and AMD messages, if there are no errors.

##### Parameters

<i>canalias</i>	A <a href="#">CanAlias</a> object.
-----------------	------------------------------------

##### Returns

A boolean value indicating a successfully reserved alias (true) or failure (false).

#### 9.9.3.17 sendMessage()

```
lcc::CANGridConnect::sendMessage (
    ... )
```

Send a message on the OpenLCB bus.

##### Parameters

...	Message options. See <a href="#">OpenLCBMessage</a> for possible options.
-----	---

#### 9.9.3.18 sendOpenLCBMessage()

```
lcc::CANGridConnect::sendOpenLCBMessage (
    message )
```

Send a message on the OpenLCB bus.

**Parameters**

<i>message</i>	An <a href="#">OpenLCBMessage</a> .
----------------	-------------------------------------

**9.9.3.19 setMessageHandler()**

```
lcc::CANGridConnect::setMessageHandler (
    handler )
```

Set the message handler.

Generally called from the upper level class to gain access to incoming messages asynchronously.

**Parameters**

<i>handler</i>	The new handler procedure.
----------------	----------------------------

**Returns**

The old handler or the empty string if there was no old handler.

**9.9.3.20 updateAliasMap()**

```
lcc::CANGridConnect::updateAliasMap (
    nid ,
    alias )
```

Update the alias map with the specified Node ID and Alias.

**Parameters**

<i>nid</i>	An OpenLCB Node ID.
<i>alias</i>	A 12-bit CAN Alias.

**9.9.4 Member Data Documentation**

#### 9.9.4.1 \_timeout

```
lcc::CANGridConnect::_timeout [private]
```

Timeout flag.

#### 9.9.4.2 \_timeoutFlag

```
lcc::CANGridConnect::_timeoutFlag [private]
```

Timeout or error message received flag.

#### 9.9.4.3 aliasMap

```
lcc::CANGridConnect::aliasMap [private]
```

Alias to NID map.

#### 9.9.4.4 canheader

```
lcc::CANGridConnect::canheader [private]
```

[CANHeader](#) component.

This component is used to extract and pack fields from and to a CAN header at a CAN Header level.

#### 9.9.4.5 datagrambuffers

```
lcc::CANGridConnect::datagrambuffers [private]
```

Datagram buffers.

#### 9.9.4.6 gcmessage

```
lcc::CANGridConnect::gcmessage [private]
```

[GridConnectMessage](#) component.

This component is used to encode CAN Messages in Grid Connect Message format for transmission.

#### 9.9.4.7 gcreply

```
lcc::CANGridConnect::gcreply [private]
```

[GridConnectReply](#) component.

This component is used to decode received Grid Connect Messages into binary CAN Messages.

#### 9.9.4.8 messagebuffers

```
lcc::CANGridConnect::messagebuffers [private]
```

General message buffers (for multi frame messages)

#### 9.9.4.9 messagehandler

```
lcc::CANGridConnect::messagehandler [private]
```

Message handler.

#### 9.9.4.10 mtidetail

```
lcc::CANGridConnect::mtidetail [private]
```

[MTIDetail](#) component.

This component is used to extract and pack fields from and to a CAN header at a MTI detail level

#### 9.9.4.11 mtiheader

```
lcc::CANGridConnect::mtiheader [private]
```

[MTIHeader](#) component.

This component is used to extract and pack fields from and to a CAN header at a MTI header level.

#### 9.9.4.12 mycanalias

```
lcc::CANGridConnect::mycanalias [private]
```

My [CanAlias](#) component.



#### 9.9.4.13 nidMap

```
lcc::CANGridConnect::nidMap [private]
```

NID to alias map.

#### 9.9.4.14 NIDPATTERN

```
lcc::CANGridConnect::NIDPATTERN [static], [private]
```

The regexp for breaking up the Node ID into bytes.

#### 9.9.4.15 parent

```
lcc::CANGridConnect::parent [private]
```

Parent instance.

#### 9.9.4.16 simplenodeflags

```
lcc::CANGridConnect::simplenodeflags [private]
```

Simple node info flags.

## 9.10 lcc::CANGridConnectOverCANSocket Class Reference

Connect to a CAN bus using GridConnect formatted message over a CAN Socket connection.

### Public Member Functions

- [CANGridConnectOverCANSocket](#) (name,...)

*Constructor: create a connection to a Grid Connect USB serial device.*

## Static Public Member Functions

- static [buildSocketnamenidDialog](#) ()  
*Function to construct the Dialog to ask the user for a CAN socket name and Node ID.*
- static [\\_CancelOpenTransport](#) ()  
*Function bound to the `Cancel` button.*
- static [\\_OpenTransport](#) ()  
*Function bound to the `Open` button.*
- static [requiredOpts](#) ()  
*Return the default option list.*
- static [drawOptionsDialog](#) (...)  
*Pop up the Options Dialog box.*

## Static Public Attributes

- static [socketnamenidDialog](#)  
*Dialog to ask the user for a socket name and Node ID.*
- static [socketnameLEntry](#)  
*[LabelSpinBox](#) containing the socket name.*
- static [nidLEntry](#)  
*[LabelEntry](#) containing the Node ID.*

## Private Attributes

- [gccomponent](#)  
*GC Component.*
- [socket](#)  
*The CAN socket.*

### 9.10.1 Detailed Description

Connect to a CAN bus using GridConnect formatted message over a CAN Socket connection.

Options:

- `-socketname` The name of the CAN Socket Device. The default is `can0`. This is a readonly option only processed at instance creation.
- `-nid` The Node ID that the computer will assume in the format of `hh:hh:hh:hh:hh:hh` which is a 48 bit number expressed as 6 pairs of hexadecimal digits separated by colons (:).
- `-promiscuousmode` Promiscuous mode flag. If true all messages are handled, whether they are addressed to this node or not.

## 9.10.2 Constructor & Destructor Documentation

### 9.10.2.1 CANGridConnectOverCANSocket()

```
lcc::CANGridConnectOverCANSocket::CANGridConnectOverCANSocket (
    name ,
    ... )
```

Constructor: create a connection to a Grid Connect USB serial device.

Connect to the CAN bus via a Grid Connect USB serial port interface.

#### Parameters

<i>name</i>	The name of the instance.
...	<p>The options:</p> <ul style="list-style-type: none"> <li>• <code>-socketname</code> The Tcp/lp port number. The default is <code>can0</code>.</li> <li>• <code>-nid</code> The Node ID that the computer will assume in the format of <code>hh:hh:hh:hh:hh:hh</code> which is a 48 bit number expressed as 6 pairs of hexadecimal digits separated by colons (:).</li> <li>• <code>-eventhandler</code> This is a script prefix that is run on incoming messages. The current message as a binary <a href="#">CanMessage</a> is appended.</li> <li>• <code>-promiscuousmode</code> Promiscuous mode flag. If true all messages are handled, whether they are addressed to this node or not.</li> </ul>

## 9.10.3 Member Function Documentation

### 9.10.3.1 \_CancelOpenTransport()

```
static lcc::CANGridConnectOverCANSocket::_CancelOpenTransport ( ) [static]
```

Function bound to the `Cancel` button.

Closes the dialog box and returns the empty string.

#### Returns

The empty string.

### 9.10.3.2 \_OpenTransport()

```
static lcc::CANGridConnectOverCANSocket::_OpenTransport ( ) [static]
```

Function bound to the `Open` button.

Closes the dialog box and returns the options needed to open the transport.

#### Returns

An option argument list with the `-nid` and `-port` options.

### 9.10.3.3 buildSocketnamenidDialog()

```
static lcc::CANGridConnectOverCANSocket::buildSocketnamenidDialog ( ) [static]
```

Function to construct the Dialog to ask the user for a CAN socket name and Node ID.

#### Returns

The Dialog box object.

### 9.10.3.4 drawOptionsDialog()

```
static lcc::CANGridConnectOverCANSocket::drawOptionsDialog (
    ... ) [static]
```

Pop up the Options Dialog box.

Pops up the Options Dialog box and collects the options needed to open the [OpenLCBOverTcp](#) object.

#### Parameters

...	<p>Options:</p> <ul style="list-style-type: none"><li>• <code>-parent</code> Set the parent for this dialog box.</li><li>• <code>-socketname</code> The default CAN Socket Name option.</li><li>• <code>-nid</code> The default Node ID to use for the Node ID option.</li></ul>
-----	--

**Returns**

Either the null string or an options list.

**9.10.3.5 requiredOpts()**

```
static lcc::CANGridConnectOverCANSocket::requiredOpts ( ) [static]
```

Return the default option list.

Returns the default options for the options dialog.

**Returns**

The option value list.

**9.10.4 Member Data Documentation****9.10.4.1 gccomponent**

```
lcc::CANGridConnectOverCANSocket::gccomponent [private]
```

GC Component.

**9.10.4.2 nidLEntry**

```
lcc::CANGridConnectOverCANSocket::nidLEntry [static]
```

LabelEntry containing the Node ID.

**9.10.4.3 socket**

```
lcc::CANGridConnectOverCANSocket::socket [private]
```

The CAN socket.

#### 9.10.4.4 socketnameLEntry

`lcc::CANGridConnectOverCANSocket::socketnameLEntry` [static]

[LabelSpinBox](#) containing the socket name.

#### 9.10.4.5 socketnamenidDialog

`lcc::CANGridConnectOverCANSocket::socketnamenidDialog` [static]

Dialog to ask the user for a socket name and Node ID.

### 9.11 lcc::CANGridConnectOverTcp Class Reference

Connect to a CAN bus using GridConnect formatted message over a Tcp/Ip connection.

#### Public Member Functions

- [CANGridConnectOverTcp](#) (name,...)  
*Constructor: create a connection to a Grid Connect USB serial device.*

#### Static Public Member Functions

- static [buildPortnidandhostDialog](#) ()  
*Function to construct the Dialog to ask the user for a port, host, and Node ID.*
- static [\\_CancelOpenTransport](#) ()  
*Function bound to the Cancel button.*
- static [\\_OpenTransport](#) ()  
*Function bound to the Open button.*
- static [requiredOpts](#) ()  
*Return the default option list.*
- static [drawOptionsDialog](#) (...)  
*Pop up the Options Dialog box.*

#### Static Public Attributes

- static [portnidandhostDialog](#)  
*Dialog to ask the user for a port, host, and Node ID.*
- static [portLSpin](#)  
*[LabelSpinBox](#) containing possible network ports.*
- static [hostLEntry](#)  
*[LabelEntry](#) containing the hostname.*
- static [nidLEntry](#)  
*[LabelEntry](#) containing the Node ID.*

## Private Attributes

- [gccomponent](#)  
*GC Component.*
- [socket](#)  
*The Tcp/Ip socket.*

### 9.11.1 Detailed Description

Connect to a CAN bus using GridConnect formatted message over a Tcp/Ip connection.

Options:

- -port The Tcp/Ip port. The default is 12021. This is a readonly option only processed at instance creation.
- -host The host name to connect to. The default is localhost. This is a readonly option only processed at instance creation.
- -nid The Node ID that the computer will assume in the format of `hh:hh:hh:hh:hh:hh` which is a 48 bit number expressed as 6 pairs of hexadecimal digits separated by colons (:).
- -promiscuousmode Promiscuous mode flag. If true all messages are handled, whether they are addressed to this node or not.

### 9.11.2 Constructor & Destructor Documentation

#### 9.11.2.1 CANGridConnectOverTcp()

```
lcc::CANGridConnectOverTcp::CANGridConnectOverTcp (
    name ,
    ... )
```

Constructor: create a connection to a Grid Connect USB serial device.

Connect to the CAN bus via a Grid Connect USB serial port interface.

#### Parameters

<i>name</i>	The name of the instance.
...	<p>The options:</p> <ul style="list-style-type: none"> <li>• -port The Tcp/Ip port number. The default is 12021.</li> <li>• -host The host to connect to. The default is localhost.</li> <li>• -nid The Node ID that the computer will assume in the format of <code>hh:hh:hh:hh:hh:hh</code> which is a 48 bit number expressed as 6 pairs of hexadecimal digits separated by colons (:).</li> <li>• -eventhandler This is a script prefix that is run on incoming messages. The current message as a binary <a href="#">CanMessage</a> is appended.</li> </ul>

### 9.11.3 Member Function Documentation

#### 9.11.3.1 `_CancelOpenTransport()`

```
static lcc::CANGridConnectOverTcp::_CancelOpenTransport ( ) [static]
```

Function bound to the `Cancel` button.

Closes the dialog box and returns the empty string.

##### Returns

The empty string.

#### 9.11.3.2 `_OpenTransport()`

```
static lcc::CANGridConnectOverTcp::_OpenTransport ( ) [static]
```

Function bound to the `Open` button.

Closes the dialog box and returns the options needed to open the transport.

##### Returns

An option argument list with the `-nid` and `-port` options.

#### 9.11.3.3 `buildPortnidandhostDialog()`

```
static lcc::CANGridConnectOverTcp::buildPortnidandhostDialog ( ) [static]
```

Function to construct the Dialog to ask the user for a port, host, and Node ID.

##### Returns

The Dialog box object.

#### 9.11.3.4 `drawOptionsDialog()`

```
static lcc::CANGridConnectOverTcp::drawOptionsDialog (
    ... ) [static]
```

Pop up the Options Dialog box.

Pops up the Options Dialog box and collects the options needed to open the [OpenLCBOverTcp](#) object.



**Parameters**

...	Options: <ul style="list-style-type: none"> <li>• -parent Set the parent for this dialog box.</li> <li>• -port The default Tcp/Ip port number option.</li> <li>• -host The default Tcp/Ip hostname option.</li> <li>• -nid The default Node ID to use for the Node ID option.</li> </ul>
-----	--

**Returns**

Either the null string or an options list.

**9.11.3.5 requiredOpts()**

```
static lcc::CANGridConnectOverTcp::requiredOpts ( ) [static]
```

Return the default option list.

Returns the default options for the options dialog.

**Returns**

The option value list.

**9.11.4 Member Data Documentation****9.11.4.1 gccomponent**

```
lcc::CANGridConnectOverTcp::gccomponent [private]
```

GC Component.

#### 9.11.4.2 hostLEntry

```
lcc::CANGridConnectOverTcp::hostLEntry [static]
```

LabelEntry containing the hostname.

#### 9.11.4.3 nidLEntry

```
lcc::CANGridConnectOverTcp::nidLEntry [static]
```

LabelEntry containing the Node ID.

#### 9.11.4.4 portLSpin

```
lcc::CANGridConnectOverTcp::portLSpin [static]
```

[LabelSpinBox](#) containing possible network ports.

#### 9.11.4.5 portnidandhostDialog

```
lcc::CANGridConnectOverTcp::portnidandhostDialog [static]
```

Dialog to ask the user for a port, host, and Node ID.

#### 9.11.4.6 socket

```
lcc::CANGridConnectOverTcp::socket [private]
```

The Tcp/Ip socket.

### 9.12 lcc::CANGridConnectOverUSBSerial Class Reference

Connect to a CAN bus using GridConnect formatted message over a USB Serial port.

## Public Member Functions

- [CANGridConnectOverUSBSerial](#) (name,...)  
*Constructor: create a connection to Grid Connect USB serial device.*

## Static Public Member Functions

- static [findAvailableComPorts](#) ()  
*Return a list of available (USB) serial ports.*
- static [buildPortandnidDialog](#) ()  
*Function to construct the Dialog to ask the user for a port and Node ID.*
- static [\\_CancelOpenTransport](#) ()  
*Function bound to the *Cancel* button.*
- static [\\_OpenTransport](#) ()  
*Function bound to the *Open* button.*
- static [requiredOpts](#) ()  
*Return the default option list.*
- static [drawOptionsDialog](#) (...)  
*Pop up the Options Dialog box.*

## Static Public Attributes

- static [portandnidDialog](#)  
*Dialog to ask the user for a port and Node ID.*
- static [portLCombo](#)  
*[LabelComboBox](#) containing all possible serial port devices.*
- static [nidLEntry](#)  
*LabelEntry containing the Node ID.*

## Private Attributes

- [gccomponent](#)  
*GC Component.*
- [ttyfd](#)  
*The tty I/O channel.*

### 9.12.1 Detailed Description

Connect to a CAN bus using GridConnect formatted message over a USB Serial port.

Options:

- -port The name of the serial port. Typically `"/dev/ttyACMn"` under Linux (using the `cdc_acm` driver). This is a readonly option only processed at instance creation.
- -nid The Node ID that the computer will assume in the format of `hh:hh:hh:hh:hh:hh` which is a 48 bit number expressed as 6 pairs of hexadecimal digits separated by colons (:).
- -promiscuousmode Promiscuous mode flag. If true all messages are handled, whether they are addressed to this node or not.

## 9.12.2 Constructor & Destructor Documentation

### 9.12.2.1 CANGridConnectOverUSBSerial()

```
lcc::CANGridConnectOverUSBSerial::CANGridConnectOverUSBSerial (
    name ,
    ... )
```

Constructor: create a connection to Grid Connect USB serial device.

Connect to the CAN bus via a Grid Connect USB serial port interface.

#### Parameters

<i>name</i>	The name of the instance.
...	<p>The options:</p> <ul style="list-style-type: none"> <li>-port The name of the serial port. Typically "/dev/ttyACMn" under Linux (using the cdc_acm driver).</li> <li>-nid The Node ID that the computer will assume in the format of <code>hh:hh:hh:hh:hh:hh</code> which is a 48 bit number expressed as 6 pairs of hexadecimal digits separated by colons (:).</li> <li>-eventhandler This is a script prefix that is run on incoming messages. The current message as a binary <a href="#">CanMessage</a> is appended.</li> <li>-promiscuousmode Promiscuous mode flag. If true all messages are handled, whether they are addressed to this node or not.</li> </ul>

## 9.12.3 Member Function Documentation

### 9.12.3.1 \_CancelOpenTransport()

```
static lcc::CANGridConnectOverUSBSerial::_CancelOpenTransport ( ) [static]
```

Function bound to the `Cancel` button.

Closes the dialog box and returns the empty string.

#### Returns

The empty string.

### 9.12.3.2 \_OpenTransport()

```
static lcc::CANGridConnectOverUSBSerial::_OpenTransport ( ) [static]
```

Function bound to the `Open` button.

Closes the dialog box and returns the options needed to open the transport.

#### Returns

An option argument list with the `-nid` and `-port` options.

### 9.12.3.3 buildPortandnidDialog()

```
static lcc::CANGridConnectOverUSBSerial::buildPortandnidDialog ( ) [static]
```

Function to construct the Dialog to ask the user for a port and Node ID.

#### Returns

The Dialog box object.

### 9.12.3.4 drawOptionsDialog()

```
static lcc::CANGridConnectOverUSBSerial::drawOptionsDialog (
    ... ) [static]
```

Pop up the Options Dialog box.

Pops up the Options Dialog box and collects the options needed to open the [CANGridConnectOverUSBSerial](#) object.

#### Parameters

...	<p>Options:</p> <ul style="list-style-type: none"><li>• <code>-parent</code> Set the parent for this dialog box.</li><li>• <code>-port</code> The default serial port name for the serial port option.</li><li>• <code>-nid</code> The default Node ID to use for the Node ID option.</li></ul>
-----	---

**Returns**

Either the null string or an options list.

**9.12.3.5 findAvailableComPorts()**

```
static lcc::CANGridConnectOverUSBSerial::findAvailableComPorts ( ) [static]
```

Return a list of available (USB) serial ports.

This method does a platform specific search for possible serial ports to use to communicate over the CAN bus.

**Returns**

A list of serial port device names.

**9.12.3.6 requiredOpts()**

```
static lcc::CANGridConnectOverUSBSerial::requiredOpts ( ) [static]
```

Return the default option list.

Returns the default options for the options dialog.

**Returns**

The option value list.

**9.12.4 Member Data Documentation****9.12.4.1 gccomponent**

```
lcc::CANGridConnectOverUSBSerial::gccomponent [private]
```

GC Component.

#### 9.12.4.2 nidLEntry

```
lcc::CANGridConnectOverUSBSerial::nidLEntry [static]
```

LabelEntry containing the Node ID.

#### 9.12.4.3 portandnidDialog

```
lcc::CANGridConnectOverUSBSerial::portandnidDialog [static]
```

Dialog to ask the user for a port and Node ID.

#### 9.12.4.4 portLCombo

```
lcc::CANGridConnectOverUSBSerial::portLCombo [static]
```

[LabelComboBox](#) containing all possible serial port devices.

#### 9.12.4.5 ttyfd

```
lcc::CANGridConnectOverUSBSerial::ttyfd [private]
```

The tty I/O channel.

## 9.13 lcc::CANHeader Class Reference

CAN Header type.

### Public Member Functions

- [CANHeader](#) (name,...)  
*Constructor: create a 29-bit CAN header.*
- [getHeader](#) ()  
*Generate and return the 29-bit header.*
- [setHeader](#) (header)  
*Decode a 29-bit CAN header.*

## Static Private Attributes

- static [RESERVED\\_SHIFT](#)  
*Bit 28 is reserved and always 1.*
- static [OPENLCBFRAME\\_SHIFT](#)  
*Bit 27 is the OpenLCB bit: 1 == OpenLCB, 0 == other CAN.*
- static [OPENLCBFRAME\\_MASK](#)  
*Bit 27 is the OpenLCB bit: 1 == OpenLCB, 0 == other CAN.*
- static [VARIABLEFIELD\\_SHIFT](#)  
*Bits 12-26 are the variable field.*
- static [VARIABLEFIELD\\_MASK](#)  
*Bits 12-26 are the variable field.*
- static [SRCID\\_SHIFT](#)  
*Bits 0-11 are the source id.*
- static [SRCID\\_MASK](#)  
*Bits 0-11 are the source id.*

### 9.13.1 Detailed Description

CAN Header type.

Creates a 29-bit CAN header. The header is generated and decoded "on the fly" from/to the supplied options:

- -openlcbframe A boolean flag to indicate an OpenLCB or generic CAN frame.
- -variablefield A 15 bit data field.
- -srcid A 12 bit source id field.

### 9.13.2 Constructor & Destructor Documentation

#### 9.13.2.1 CANHeader()

```
lcc::CANHeader::CANHeader (
    name ,
    ... )
```

Constructor: create a 29-bit CAN header.

Creates a CAN header object from the supplied options.

#### Parameters

<i>name</i>	The name of the object.	Generated by Doxygen
...	Options:	
	<ul style="list-style-type: none"> <li>• -openlcbframe Flag to indicate a OpenLCB frame or not. Default yes, type boolean.</li> <li>• -variablefield Fifteen bit variable field. Default 0, type 15-bit integer.</li> <li>• -srcid Twelve bit source id field. Default 0, type 12-bit integer.</li> </ul>	



### 9.13.3 Member Function Documentation

#### 9.13.3.1 getHeader()

```
lcc::CANHeader::getHeader ( )
```

Generate and return the 29-bit header.

Creates a 29-bit header from the supplied options.

#### Returns

The 29-bit CAN header.

#### 9.13.3.2 setHeader()

```
lcc::CANHeader::setHeader (
    header )
```

Decode a 29-bit CAN header.

The 29-bit CAN header is decoded and the various options set.

#### Parameters

<i>header</i>	The 29-bit CAN header.
---------------	------------------------

### 9.13.4 Member Data Documentation

#### 9.13.4.1 OPENLCBFRAME\_MASK

```
lcc::CANHeader::OPENLCBFRAME_MASK [static], [private]
```

Bit 27 is the OpenLCB bit: 1 == OpenLCB, 0 == other CAN.

#### 9.13.4.2 OPENLCBFRAME\_SHIFT

```
lcc::CANHeader::OPENLCBFRAME_SHIFT [static], [private]
```

Bit 27 is the OpenLCB bit: 1 == OpenLCB, 0 == other CAN.

#### 9.13.4.3 RESERVED\_SHIFT

```
lcc::CANHeader::RESERVED_SHIFT [static], [private]
```

Bit 28 is reserved and always 1.

#### 9.13.4.4 SRCID\_MASK

```
lcc::CANHeader::SRCID_MASK [static], [private]
```

Bits 0-11 are the source id.

#### 9.13.4.5 SRCID\_SHIFT

```
lcc::CANHeader::SRCID_SHIFT [static], [private]
```

Bits 0-11 are the source id.

#### 9.13.4.6 VARIABLEFIELD\_MASK

```
lcc::CANHeader::VARIABLEFIELD_MASK [static], [private]
```

Bits 12-26 are the variable field.

#### 9.13.4.7 VARIABLEFIELD\_SHIFT

```
lcc::CANHeader::VARIABLEFIELD_SHIFT [static], [private]
```

Bits 12-26 are the variable field.

## 9.14 Icc::CanMessage Class Reference

A CAN Message, containing a 29-bit header and upto 8 bytes of data.

### Public Member Functions

- [CanMessage](#) (name,...)  
*Constructor: create a CANMessage object Creates a fresh CANMessage object, with possible initialization.*
- [hashCode](#) ()  
*Return a hash code.*
- [equals](#) (a)  
*Equality check.*
- [replyExpected](#) ()  
*Reply expected.*
- [setNumDataElements](#) (n)  
*Set the number of data elements.*
- [setData](#) (d)  
*Set the data values.*
- [getData](#) ()  
*Return the data vector.*
- [getHeader](#) ()  
*Return the header.*
- [setHeader](#) (h)  
*Set the header.*
- [toString](#) ()  
*Method to create a string version of the message.*

### Static Public Member Functions

- static [copy](#) (m)  
*Copy constructor.*
- static [validate](#) (o)  
*Validator typemethod.*

### Public Attributes

- [\\_header](#)  
*The header.*

### 9.14.1 Detailed Description

A CAN Message, containing a 29-bit header and upto 8 bytes of data.

Options:

- -header The 29-bit header. Readonly, used only during creation. Default 0.
- -length The length of the data. Readonly, used only during creation. Default 0.
- -data The initial data. Readonly, used only during creation. Default is the empty list.
- -extended. Boolean flag to indicate an extended protocol frame. Default is false.
- -rtr. Boolean flag to indicate if a reply is expected. Default is false.

Additional methods defined using the macros AbstractMessage and AbstractMRMessage include:

- getElement {n} – Get the nth data element.
- getNumDataElements {} – Get the number of data elements.
- setElement {n v} – Set the nth data element.
- setOpCode {i} – Set the opcode (byte 0).
- getOpCode {} – Get the opcode (byte 0).
- getOpCodeHex {} – Get the opcode (byte 0) in hex.
- setNeededMode {pMode} – Set the needed mode.
- getNeededMode {} – Get the needed mode.
- replyExpected {} – Returns reply expected flag.
- isBinary {} – Returns binary flag.
- setBinary {b} – Set the binary flag.
- setTimeout {t} – Set the timeout.
- getTimeout {} – Get the timeout.
- setRetries {i} – Set the number of retries.
- getRetries {} – Get the number of retries.
- addIntAsThree {val offset} – Insert an integer as three decimal digits (with leading 0s).
- addIntAsTwoHex {val offset} – Insert an integer as two hexadecimal digits (with leading 0s).
- addIntAsThreeHex {val offset} – Insert an integer as three hexadecimal digits (with leading 0s).
- addIntAsFourHex {val offset} – Insert an integer as four hexadecimal digits (with leading 0s).
- setNumDataElements {n} – Set the number of data bytes.
- toString {} – Return the data object as a string.

And these (private) instance variables:

- `_dataChars` {}
- `_nDataChars` 0
- `mNeededMode` 0
- `_isBinary` false
- `mTimeout` 0
- `mRetries` 0

And these (private) static variables:

- `SHORT_TIMEOUT` 2000
- `LONG_TIMEOUT` 60000

## 9.14.2 Constructor & Destructor Documentation

### 9.14.2.1 CanMessage()

```
lcc::CanMessage::CanMessage (
    name ,
    ... )
```

Constructor: create a CANMessage object Creates a fresh CANMessage object, with possible initialization.

#### Parameters

<i>name</i>	The name of the new instance.
...	<p>The options:</p> <ul style="list-style-type: none"><li>• <code>-header</code> The 29-bit header. Readonly, used only during creation.</li><li>• <code>-length</code> The length of the data. Readonly, used only during creation.</li><li>• <code>-data</code> The initial data. Readonly, used only during creation.</li><li>• <code>-extended</code>. Boolean flag to indicate an extended protocol frame.</li><li>• <code>-rtr</code>. Boolean flag to indicate if a reply is expected.</li></ul>

### 9.14.3 Member Function Documentation

#### 9.14.3.1 `copy()`

```
static lcc::CanMessage::copy (
    m ) [static]
```

Copy constructor.

Copies a CANMessage instance.

##### Parameters

<i>m</i>	The CANMessage to make a copy of.
----------	-----------------------------------

#### 9.14.3.2 `equals()`

```
lcc::CanMessage::equals (
    a )
```

Equality check.

CANMessages are equal if all of the bits are the same.

##### Parameters

<i>a</i>	A CANMessage to compare to.
----------	-----------------------------

##### Returns

A boolean value indication equality.

#### 9.14.3.3 `getData()`

```
lcc::CanMessage::getData ( )
```

Return the data vector.

##### Returns

The data vector.

#### 9.14.3.4 getHeader()

```
lcc::CanMessage::getHeader ( )
```

Return the header.

##### Returns

The header.

#### 9.14.3.5 hashCode()

```
lcc::CanMessage::hashCode ( )
```

Return a hash code.

##### Returns

The header as the object's hash code.

#### 9.14.3.6 replyExpected()

```
lcc::CanMessage::replyExpected ( )
```

Reply expected.

##### Returns

A boolean flag indicating if a reply is expected.

#### 9.14.3.7 setData()

```
lcc::CanMessage::setData (
    d )
```

Set the data values.

Copy data into the data vector.

**Parameters**

<i>d</i>	Replacement data values.
----------	--------------------------

**9.14.3.8 setHeader()**

```
lcc::CanMessage::setHeader (
    h )
```

Set the header.

**Parameters**

<i>h</i>	The new header.
----------	-----------------

**9.14.3.9 setNumDataElements()**

```
lcc::CanMessage::setNumDataElements (
    n )
```

Set the number of data elements.

Sets the number of data elements.

**Parameters**

<i>n</i>	The number of data elements.
----------	------------------------------

**9.14.3.10 toString()**

```
lcc::CanMessage::toString ( )
```

Method to create a string version of the message.

**Returns**

A string representation of the message.



## 9.14.3.11 validate()

```
static lcc::CanMessage::validate (
    o ) [static]
```

Validator type method.

## Parameters

<i>o</i>	The object to validate.
----------	-------------------------

## 9.14.4 Member Data Documentation

## 9.14.4.1 \_header

```
lcc::CanMessage::_header
```

The header.

## 9.15 lcc::CanTransport Class Reference

Logical transport of CAN Messages.

## Public Member Functions

- [CanTransport](#) (name,...)  
*Construct a [CanTransport](#) object.*

## Private Member Functions

- [\\_transportlayerconf](#) (opt, value)  
*Set the transport layer component.*

## Private Attributes

- [transport](#)  
*Transport Layer component.*

### 9.15.1 Detailed Description

Logical transport of CAN Messages.

CAN Bus abstraction layer

Options:

- -transportlayer The physical transport layer (eg GridConnectTransport over USB serial, etc.)
- -readhandler The read handler for incoming messages.

### 9.15.2 Constructor & Destructor Documentation

#### 9.15.2.1 CanTransport()

```
lcc::CanTransport::CanTransport (
    name ,
    ... )
```

Construct a [CanTransport](#) object.

#### Parameters

<i>name</i>	The name of the transport object.
...	Options: <ul style="list-style-type: none"><li>• -transportlayer The physical transport layer (eg GridConnectTransport over USB serial, etc.)</li><li>• -readhandler The read handler for incoming messages.</li></ul>

### 9.15.3 Member Function Documentation

## 9.15.3.1 \_transportlayerconf()

```
lcc::CanTransport::_transportlayerconf (
    opt ,
    value ) [private]
```

Set the transport layer component.

## Parameters

<i>opt</i>	Always -transportlayer
<i>value</i>	The physical transport object.

## 9.15.4 Member Data Documentation

## 9.15.4.1 transport

```
lcc::CanTransport::transport [private]
```

Transport Layer component.

## 9.16 FCFSupport::Car Class Reference

This class holds all of the information for a single car.

```
#include <Car.h>
```

## Public Member Functions

- [Car](#) ()  
*Default constructor.*
- [Car](#) ([Car](#) &other)  
*Copy constructor.*
- [Car](#) & [operator=](#) ([Car](#) &other)  
*Assignment operator.*
- [Car](#) (char t, const char \*m, const char \*n, const char \*d, int l, int p, int wc, int lw, int ldw, bool lp, bool mp, bool fp, const [Owner](#) \*own, bool dp, const [Train](#) \*lt, int mvs, [Industry](#) \*loc, [Industry](#) \*dest, int trps, int asgns)  
*Full constructor.*
- char [Type](#) () const  
*Return the car type.*
- void [SetType](#) (char t)



- void [SetFixedRouteP](#) (bool f)  
*Set whether this car is on a fixed route.*
- const [Owner](#) \* [CarOwner](#) () const  
*Return the car's owner.*
- void [SetCarOwner](#) (const [Owner](#) \*o)  
*Set the car's owner.*
- bool [IsDoneP](#) () const  
*Is this car done?*
- void [SetDone](#) ()  
*Flag this car as done.*
- void [SetNotDone](#) ()  
*Flag this car as not done.*
- const [Train](#) \* [LastTrain](#) () const  
*Return the last train to move this car.*
- void [SetLastTrain](#) (const [Train](#) \*lt)  
*Set the last train to move this car.*
- const [Train](#) \* [PrevTrain](#) () const  
*Return the previous train to move this car.*
- void [SetPrevTrain](#) (const [Train](#) \*lt)  
*Set the previous train to move this car.*
- int [MovementsThisSession](#) () const  
*Return the number of movements this session.*
- void [ClearMovementsThisSession](#) ()  
*Clear the number of movements this session.*
- void [IncrmentMovementsThisSession](#) ()  
*Increment the number of movements this session.*
- [Industry](#) \* [Location](#) () const  
*Return the location of this car.*
- void [SetLocation](#) ([Industry](#) \*newloc)  
*Set the location of this car.*
- [Industry](#) \* [Destination](#) () const  
*Return the destination of this car.*
- void [SetDestination](#) ([Industry](#) \*newdest)  
*Set the destination of this car.*
- int [Trips](#) () const  
*Return the number of trips this car has had.*
- void [ClearTrips](#) ()  
*Clear the number of trips this car has had.*
- void [IncrementTrips](#) ()  
*Increment the number of trips this car has had.*
- int [Assignments](#) () const  
*Return the number of assignments this car has had.*
- void [SetAssignments](#) (int a)  
*Set the number of assignments this car has had.*
- void [ClearAssignments](#) ()  
*Clear the number of assignments this car has had.*
- void [IncrementAssignments](#) ()

*Increment the number of assignments this car has had.*

- bool [Peek](#) () const

*Return the peek flag.*

- void [SetPeek](#) (bool p=false)

*Set or clear the peek flag.*

## Private Attributes

- const [Owner](#) \* [owner](#)

*The owner of this car.*

- const [Train](#) \* [lasttrain](#)

*The last train to handle this car.*

- const [Train](#) \* [prevtrain](#)

*The previous train to handle this car.*

- [Industry](#) \* [location](#)

*This car's location.*

- [Industry](#) \* [destination](#)

*This car's destination.*

- string [marks](#)

*This car's reporting marks.*

- string [number](#)

*This car's number.*

- string [divisions](#)

*This car's division list.*

- int [length](#)

*This car's length.*

- int [plate](#)

*This car's clearance plate.*

- int [weightclass](#)

*This car's weight class.*

- int [ltwt](#)

*This car's empty weight.*

- int [ldlmt](#)

*This car's loaded weight.*

- int [trips](#)

*The number of trips this car has made.*

- int [moves](#)

*The number of moves this car has made.*

- int [assignments](#)

*The number of assignments this car has had.*

- bool [loadedP](#)

*This car's loaded flag.*

- bool [mirrorP](#)

*This car's mirror flag.*

- bool [fixedP](#)

*This car's fixed route flag.*

- bool `doneP`  
*This car's done flag.*
- bool `peek`  
*This car's peel flak.*
- bool `tmpStatus`  
*Temp status flag.*
- char `type`  
*This car's type.*

## Friends

- class `System`  
*The `System` class is a friend.*

### 9.16.1 Detailed Description

This class holds all of the information for a single car.

Including its reporting marks, car number, type, division list, owner, length, weight, and so on.

```
@author Robert Heller \<heller\@deepsoft.com\>
```

### 9.16.2 Constructor & Destructor Documentation

#### 9.16.2.1 `Car()` [1/3]

```
FCFSupport::Car::Car ( ) [inline]
```

Default constructor.

All slots are initialized to default values.

References assignments, destination, divisions, `doneP`, `fixedP`, `lasttrain`, `ldlmt`, `length`, `loadedP`, `location`, `ltwt`, `marks`, `mirrorP`, `moves`, `number`, `owner`, `peek`, `plate`, `prevtrain`, `tmpStatus`, `trips`, `type`, and `weightclass`.

#### 9.16.2.2 `Car()` [2/3]

```
FCFSupport::Car::Car (
    Car & other ) [inline]
```

Copy constructor.

All slots are copied.

## Parameters

<i>other</i>	The originating instance.
--------------	---------------------------

References assignments, destination, divisions, doneP, fixedP, lasttrain, ldlmt, length, loadedP, location, ltwt, marks, mirrorP, moves, number, owner, peek, plate, prevtrain, tmpStatus, trips, type, and weightclass.

## 9.16.2.3 Car() [3/3]

```
FCFSupport::Car::Car (
    char t,
    const char * m,
    const char * n,
    const char * d,
    int l,
    int p,
    int wc,
    int lw,
    int ldw,
    bool lp,
    bool mp,
    bool fp,
    const Owner * own,
    bool dp,
    const Train * lt,
    int mvs,
    Industry * loc,
    Industry * dest,
    int trps,
    int asgns ) [inline]
```

Full constructor.

Fill all slots from the argument list.

## Parameters

<i>t</i>	<a href="#">Car</a> type.
<i>m</i>	Reporting marks (railroad).
<i>n</i>	Number.
<i>d</i>	<a href="#">Division</a> symbol list.
<i>l</i>	Length.
<i>p</i>	Plate.
<i>wc</i>	Weight class.
<i>lw</i>	Light (empty) weight.
<i>ldw</i>	Load limit (loaded weight).
<i>lp</i>	Is the car loaded?



## Parameters

<i>mp</i>	Can the car be mirrored?
<i>fp</i>	Does it have a fixed route?
<i>own</i>	<a href="#">Car</a> owner.
<i>dp</i>	Is it done moving?
<i>lt</i>	The last train to handle this car.
<i>mvs</i>	The number of times this car has been moved this session.
<i>loc</i>	The car's current location.
<i>dest</i>	The car's destination.
<i>trps</i>	The number of trips this car has made.
<i>asgns</i>	The number of times this car has been assigned.

References assignments, destination, divisions, doneP, fixedP, lasttrain, ldlmt, length, loadedP, location, ltwt, marks, mirrorP, moves, number, owner, plate, prevtrain, trips, type, and weightclass.

### 9.16.3 Member Function Documentation

#### 9.16.3.1 Assignments()

```
int FCFSupport::Car::Assignments ( ) const [inline]
```

Return the number of assignments this car has had.

References assignments.

#### 9.16.3.2 CarOwner()

```
const Owner* FCFSupport::Car::CarOwner ( ) const [inline]
```

Return the car's owner.

References owner.

#### 9.16.3.3 ClearAssignments()

```
void FCFSupport::Car::ClearAssignments ( ) [inline]
```

Clear the number of assignments this car has had.

References assignments.

#### 9.16.3.4 ClearMovementsThisSession()

```
void FCFSupport::Car::ClearMovementsThisSession ( ) [inline]
```

Clear the number of movements this session.

References moves.

#### 9.16.3.5 ClearTrips()

```
void FCFSupport::Car::ClearTrips ( ) [inline]
```

Clear the number of trips this car has had.

References trips.

#### 9.16.3.6 Destination()

```
Industry* FCFSupport::Car::Destination ( ) const [inline]
```

Return the destination of this car.

References destination.

#### 9.16.3.7 Divisions()

```
const char* FCFSupport::Car::Divisions ( ) const [inline]
```

Return the car's division list.

References divisions, and SetDivisions().

#### 9.16.3.8 EmptyP()

```
bool FCFSupport::Car::EmptyP ( ) const [inline]
```

Is the car empty?

References loadedP.

#### 9.16.3.9 FixedRouteP()

```
bool FCFSupport::Car::FixedRouteP ( ) const [inline]
```

Is this car on a fixed route?

References fixedP.

#### 9.16.3.10 IncrementAssignments()

```
void FCFSupport::Car::IncrementAssignments ( ) [inline]
```

Increment the number of assignments this car has had.

References assignments.

#### 9.16.3.11 IncrementTrips()

```
void FCFSupport::Car::IncrementTrips ( ) [inline]
```

Increment the number of trips this car has had.

References trips.

#### 9.16.3.12 IncrmentMovementsThisSession()

```
void FCFSupport::Car::IncrmentMovementsThisSession ( ) [inline]
```

Increment the number of movements this session.

References moves.

#### 9.16.3.13 IsDoneP()

```
bool FCFSupport::Car::IsDoneP ( ) const [inline]
```

Is this car done?

References doneP.

**9.16.3.14 LastTrain()**

```
const Train* FCFSupport::Car::LastTrain ( ) const [inline]
```

Return the last train to move this car.

References lasttrain.

**9.16.3.15 LdLmt()**

```
int FCFSupport::Car::LdLmt ( ) const [inline]
```

Return the car's load limit.

References ldlmt.

**9.16.3.16 Length()**

```
int FCFSupport::Car::Length ( ) const [inline]
```

Return the car's length.

References length.

**9.16.3.17 Load()**

```
void FCFSupport::Car::Load ( ) [inline]
```

Load the car.

References loadedP.

**9.16.3.18 LoadedP()**

```
bool FCFSupport::Car::LoadedP ( ) const [inline]
```

Is the car loaded?

References loadedP.

#### 9.16.3.19 Location()

```
Industry* FCFSupport::Car::Location ( ) const [inline]
```

Return the location of this car.

References location.

#### 9.16.3.20 LtWt()

```
int FCFSupport::Car::LtWt ( ) const [inline]
```

Return the car's empty weight.

References ltwt.

#### 9.16.3.21 Marks()

```
const char* FCFSupport::Car::Marks ( ) const [inline]
```

Return the car's reporting marks (railroad).

References marks, and SetMarks().

#### 9.16.3.22 MovementsThisSession()

```
int FCFSupport::Car::MovementsThisSession ( ) const [inline]
```

Return the number of movements this session.

References moves.

#### 9.16.3.23 Number()

```
const char* FCFSupport::Car::Number ( ) const [inline]
```

Return the car's number.

References number, and SetNumber().

#### 9.16.3.24 OkToMirrorP()

```
bool FCFSupport::Car::OkToMirrorP ( ) const [inline]
```

Is it OK to mirror this car?

References mirrorP.

#### 9.16.3.25 operator=()

```
Car& FCFSupport::Car::operator= (
    Car & other ) [inline]
```

Assignment operator.

All slots are copied.

##### Parameters

<i>other</i>	The right hand operand.
--------------	-------------------------

References assignments, destination, divisions, doneP, fixedP, lasttrain, ldlmt, length, loadedP, location, ltwt, marks, mirrorP, moves, number, owner, peek, plate, prevtrain, tmpStatus, trips, type, and weightclass.

#### 9.16.3.26 Peek()

```
bool FCFSupport::Car::Peek ( ) const [inline]
```

Return the peek flag.

References peek.

#### 9.16.3.27 Plate()

```
int FCFSupport::Car::Plate ( ) const [inline]
```

Return the car's clearance plate.

References plate.

#### 9.16.3.28 PrevTrain()

```
const Train* FCFSupport::Car::PrevTrain ( ) const [inline]
```

Return the previous train to move this car.

References prevtrain.

#### 9.16.3.29 SetAssignments()

```
void FCFSupport::Car::SetAssignments (
    int a ) [inline]
```

Set the number of assignments this car has had.

References assignments.

#### 9.16.3.30 SetCarOwner()

```
void FCFSupport::Car::SetCarOwner (
    const Owner * o ) [inline]
```

Set the car's owner.

References owner.

#### 9.16.3.31 SetDestination()

```
void FCFSupport::Car::SetDestination (
    Industry * newdest ) [inline]
```

Set the destination of this car.

References destination.

#### 9.16.3.32 SetDivisions()

```
void FCFSupport::Car::SetDivisions (
    string d ) [inline]
```

Set the car's division list.

References divisions.

Referenced by Divisions().

#### 9.16.3.33 SetDone()

```
void FCFSupport::Car::SetDone ( ) [inline]
```

Flag this car as done.

References doneP.

#### 9.16.3.34 SetFixedRouteP()

```
void FCFSupport::Car::SetFixedRouteP (
    bool f ) [inline]
```

Set whether this car is on a fixed route.

References fixedP.

#### 9.16.3.35 SetLastTrain()

```
void FCFSupport::Car::SetLastTrain (
    const Train * lt ) [inline]
```

Set the last train to move this car.

References lasttrain.



#### 9.16.3.36 SetLdLmt()

```
void FCFSupport::Car::SetLdLmt (
    int ldw ) [inline]
```

Set the car's load limit.

References `ldlmt`.

#### 9.16.3.37 SetLength()

```
void FCFSupport::Car::SetLength (
    int l ) [inline]
```

Set the car's length.

References `length`.

#### 9.16.3.38 SetLocation()

```
void FCFSupport::Car::SetLocation (
    Industry * newloc ) [inline]
```

Set the location of this car.

References `location`.

#### 9.16.3.39 SetLtWt()

```
void FCFSupport::Car::SetLtWt (
    int lw ) [inline]
```

Set the car's empty weight.

References `ltwt`.

#### 9.16.3.40 SetMarks()

```
void FCFSupport::Car::SetMarks (
    string m ) [inline]
```

Set the car's reporting marks.

References marks.

Referenced by Marks().

#### 9.16.3.41 SetNotDone()

```
void FCFSupport::Car::SetNotDone ( ) [inline]
```

Flag this car as not done.

References doneP.

#### 9.16.3.42 SetNumber()

```
void FCFSupport::Car::SetNumber (
    string n ) [inline]
```

Set the car's number.

References number.

Referenced by Number().

#### 9.16.3.43 SetOkToMirrorP()

```
void FCFSupport::Car::SetOkToMirrorP (
    bool m ) [inline]
```

Set this car's mirror status.

References mirrorP.

#### 9.16.3.44 SetPeek()

```
void FCFSupport::Car::SetPeek (
    bool p = false ) [inline]
```

Set or clear the peek flag.

References peek.

#### 9.16.3.45 SetPlate()

```
void FCFSupport::Car::SetPlate (
    int p ) [inline]
```

Set the car's clearance plate.

References plate.

#### 9.16.3.46 SetPrevTrain()

```
void FCFSupport::Car::SetPrevTrain (
    const Train * lt ) [inline]
```

Set the previous train to move this car.

References prevtrain.

#### 9.16.3.47 SetType()

```
void FCFSupport::Car::SetType (
    char t ) [inline]
```

Set the car type.

##### Parameters

<i>t</i>	The new car type.
----------	-------------------

References type.

**9.16.3.48 SetWeightClass()**

```
void FCFSupport::Car::SetWeightClass (
    int wc ) [inline]
```

Set the car's weight class.

References weightclass.

**9.16.3.49 Trips()**

```
int FCFSupport::Car::Trips ( ) const [inline]
```

Return the number of trips this car has had.

References trips.

**9.16.3.50 Type()**

```
char FCFSupport::Car::Type ( ) const [inline]
```

Return the car type.

References type.

**9.16.3.51 UnLoad()**

```
void FCFSupport::Car::UnLoad ( ) [inline]
```

Unload the car.

References loadedP.

**9.16.3.52 WeightClass()**

```
int FCFSupport::Car::WeightClass ( ) const [inline]
```

Return the car's weight class.

References weightclass.

## 9.16.4 Friends And Related Function Documentation

### 9.16.4.1 System

```
friend class System [friend]
```

The [System](#) class is a friend.

## 9.16.5 Member Data Documentation

### 9.16.5.1 assignments

```
int FCFSupport::Car::assignments [private]
```

The number of assignments this car has had.

Referenced by [Assignments\(\)](#), [Car\(\)](#), [ClearAssignments\(\)](#), [IncrementAssignments\(\)](#), [operator=\(\)](#), and [SetAssignments\(\)](#).

### 9.16.5.2 destination

```
Industry\* FCFSupport::Car::destination [private]
```

This car's destination.

Referenced by [Car\(\)](#), [Destination\(\)](#), [operator=\(\)](#), and [SetDestination\(\)](#).

### 9.16.5.3 divisions

```
string FCFSupport::Car::divisions [private]
```

This car's division list.

Referenced by [Car\(\)](#), [Divisions\(\)](#), [operator=\(\)](#), and [SetDivisions\(\)](#).

#### 9.16.5.4 doneP

```
bool FCFSupport::Car::doneP [private]
```

This car's done flag.

Referenced by Car(), IsDoneP(), operator=(), SetDone(), and SetNotDone().

#### 9.16.5.5 fixedP

```
bool FCFSupport::Car::fixedP [private]
```

This car's fixed route flag.

Referenced by Car(), FixedRouteP(), operator=(), and SetFixedRouteP().

#### 9.16.5.6 lasttrain

```
const Train* FCFSupport::Car::lasttrain [private]
```

The last train to handle this car.

Referenced by Car(), LastTrain(), operator=(), and SetLastTrain().

#### 9.16.5.7 ldLmt

```
int FCFSupport::Car::ldLmt [private]
```

This car's loaded weight.

Referenced by Car(), LdLmt(), operator=(), and SetLdLmt().

#### 9.16.5.8 length

```
int FCFSupport::Car::length [private]
```

This car's length.

Referenced by Car(), Length(), operator=(), and SetLength().

#### 9.16.5.9 loadedP

```
bool FCFSupport::Car::loadedP [private]
```

This car's loaded flag.

Referenced by Car(), EmptyP(), Load(), LoadedP(), operator=(), and UnLoad().

#### 9.16.5.10 location

```
Industry* FCFSupport::Car::location [private]
```

This car's location.

Referenced by Car(), Location(), operator=(), and SetLocation().

#### 9.16.5.11 ltwt

```
int FCFSupport::Car::ltwt [private]
```

This car's empty weight.

Referenced by Car(), LtWt(), operator=(), and SetLtWt().

#### 9.16.5.12 marks

```
string FCFSupport::Car::marks [private]
```

This car's reporting marks.

Referenced by Car(), Marks(), operator=(), and SetMarks().

#### 9.16.5.13 mirrorP

```
bool FCFSupport::Car::mirrorP [private]
```

This car's mirror flag.

Referenced by Car(), OkToMirrorP(), operator=(), and SetOkToMirrorP().

#### 9.16.5.14 moves

```
int FCFSupport::Car::moves [private]
```

The number of moves this car has made.

Referenced by `Car()`, `ClearMovementsThisSession()`, `IncrmentMovementsThisSession()`, `MovementsThisSession()`, and `operator=()`.

#### 9.16.5.15 number

```
string FCFSupport::Car::number [private]
```

This car's number.

Referenced by `Car()`, `Number()`, `operator=()`, and `SetNumber()`.

#### 9.16.5.16 owner

```
const Owner* FCFSupport::Car::owner [private]
```

The owner of this car.

Referenced by `Car()`, `CarOwner()`, `operator=()`, and `SetCarOwner()`.

#### 9.16.5.17 peek

```
bool FCFSupport::Car::peek [private]
```

This car's peel flak.

Referenced by `Car()`, `operator=()`, `Peek()`, and `SetPeek()`.

#### 9.16.5.18 plate

```
int FCFSupport::Car::plate [private]
```

This car's clearance plate.

Referenced by `Car()`, `operator=()`, `Plate()`, and `SetPlate()`.



#### 9.16.5.19 prevtrain

```
const Train* FCFSupport::Car::prevtrain [private]
```

The previous train to handle this car.

Referenced by Car(), operator=(), PrevTrain(), and SetPrevTrain().

#### 9.16.5.20 tmpStatus

```
bool FCFSupport::Car::tmpStatus [private]
```

Temp status flag.

Referenced by Car(), and operator=().

#### 9.16.5.21 trips

```
int FCFSupport::Car::trips [private]
```

The number of trips this car has made.

Referenced by Car(), ClearTrips(), IncrementTrips(), operator=(), and Trips().

#### 9.16.5.22 type

```
char FCFSupport::Car::type [private]
```

This car's type.

Referenced by Car(), operator=(), SetType(), and Type().

#### 9.16.5.23 weightclass

```
int FCFSupport::Car::weightclass [private]
```

This car's weight class.

Referenced by Car(), operator=(), SetWeightClass(), and WeightClass().

## 9.17 FCFSupport::CarGroup Class Reference

Car group class.

```
#include <CarType.h>
```

### Public Types

- enum [CarGroupConsts](#) { [MaxCarGroup](#) = 16 }  
*Car group constants.*

### Public Member Functions

- [CarGroup](#) ()  
*Default constructor.*
- [CarGroup](#) ([CarGroup](#) &other)  
*Copy constructor.*
- [CarGroup](#) & [operator=](#) ([CarGroup](#) &other)  
*Assignment operator.*
- [CarGroup](#) (char g, const char \*d)  
*Full constructor.*
- char [Group](#) () const  
*Return the group code.*
- const char \* [Description](#) () const  
*Return the description string.*

### Private Attributes

- string [description](#)  
*The description string.*
- char [group](#)  
*The car group code.*

### 9.17.1 Detailed Description

Car group class.

Not presently used.

Author

Robert Heller <heller@deepsoft.com>

### 9.17.2 Member Enumeration Documentation

#### 9.17.2.1 CarGroupConsts

```
enum FCFSupport::CarGroup::CarGroupConsts
```

Car group constants.

## Enumerator

MaxCarGroup	The maximum number of car groups.
-------------	-----------------------------------

### 9.17.3 Constructor & Destructor Documentation

#### 9.17.3.1 CarGroup() [1/3]

```
FCFSupport::CarGroup::CarGroup ( ) [inline]
```

Default constructor.

Initialize all slots to me empty.

References FCFSupport::CarType::group.

#### 9.17.3.2 CarGroup() [2/3]

```
FCFSupport::CarGroup::CarGroup (
    CarGroup & other ) [inline]
```

Copy constructor.

Create a car group that is a clone of another.

## Parameters

<i>other</i>	The other car group instance.
--------------	-------------------------------

References description, FCFSupport::CarType::group, and group.

#### 9.17.3.3 CarGroup() [3/3]

```
FCFSupport::CarGroup::CarGroup (
    char g,
    const char * d ) [inline]
```

Full constructor.

**Parameters**

<i>g</i>	Car group character code.
<i>d</i>	Description of this car group.

References FCFSupport::CarType::group.

## 9.17.4 Member Function Documentation

### 9.17.4.1 Description()

```
const char* FCFSupport::CarGroup::Description ( ) const [inline]
```

Return the description string.

### 9.17.4.2 Group()

```
char FCFSupport::CarGroup::Group ( ) const [inline]
```

Return the group code.

References FCFSupport::CarType::group.

### 9.17.4.3 operator=()

```
CarGroup& FCFSupport::CarGroup::operator= (
    CarGroup & other ) [inline]
```

Assignment operator.

Create a car group that is a clone of another.

**Parameters**

<i>other</i>	The other car group instance.
--------------	-------------------------------

References description, FCFSupport::CarType::group, and group.

### 9.17.5 Member Data Documentation

#### 9.17.5.1 description

```
string FCFSupport::CarGroup::description [private]
```

The description string.

Referenced by CarGroup(), and operator=().

#### 9.17.5.2 group

```
char FCFSupport::CarGroup::group [private]
```

The car group code.

Referenced by CarGroup(), and operator=().

## 9.18 FCFSupport::CarType Class Reference

The [CarType](#) class represents a type of railroad car (rolling stock).

```
#include <CarType.h>
```

### Public Types

- enum [CarTypeConsts](#) { [NumberOfCarTypes](#) = 91, [MaxCarTypes](#) = 128 }  
*Some specific constants relating to car types.*

### Public Member Functions

- [CarType](#) ()  
*Default constructor.*
- [CarType](#) ([CarType](#) &other)  
*Copy constructor.*
- [CarType](#) & [operator=](#) ([CarType](#) &other)  
*Assignment operator.*
- [CarType](#) (const char \*c, const char \*t, char g)  
*Full constructor.*
- [~CarType](#) ()  
*Destructor.*
- const char \* [Comment](#) () const  
*Return the car type's commentary.*
- const char \* [Type](#) () const  
*Return the car type name.*
- char [Group](#) () const  
*Return the car type's group code.*

## Private Attributes

- string [comment](#)  
*The commentary string.*
- string [type](#)  
*The type name.*
- char [group](#)  
*The group code.*

## Friends

- class [System](#)  
*The [System](#) class is a friend.*

### 9.18.1 Detailed Description

The [CarType](#) class represents a type of railroad car (rolling stock).

[Car](#) types are represented as a single printable character and have associated with that printable character is a type name and possibly a short commentary.

[Car](#) types are also collected into groups as well.

```
@author Robert Heller \<heller\@deepsoft.com\>
```

### 9.18.2 Member Enumeration Documentation

#### 9.18.2.1 CarTypeConsts

```
enum FCFSupport::CarType::CarTypeConsts
```

Some specific constants relating to car types.

#### Enumerator

<a href="#">NumberOfCarTypes</a>	The number of usable car type characters.
<a href="#">MaxCarTypes</a>	The maximum number of car types (based on 7-bit ASCII).

### 9.18.3 Constructor & Destructor Documentation

#### 9.18.3.1 CarType() [1/3]

```
FCFSupport::CarType::CarType ( ) [inline]
```

Default constructor.

Create a default instance.

References comment, group, and type.

#### 9.18.3.2 CarType() [2/3]

```
FCFSupport::CarType::CarType (
    CarType & other ) [inline]
```

Copy constructor.

Copy a car type from another instance.

##### Parameters

<i>other</i>	The other instance.
--------------	---------------------

References comment, group, and type.

#### 9.18.3.3 CarType() [3/3]

```
FCFSupport::CarType::CarType (
    const char * c,
    const char * t,
    char g ) [inline]
```

Full constructor.

Create a fully qualified car type object.

**Parameters**

<i>c</i>	The name of the car type.
<i>t</i>	The brief commentary about the car type.
<i>g</i>	The car type's group code.

References comment, group, and type.

**9.18.3.4 ~CarType()**

```
FCFSupport::CarType::~~CarType ( ) [inline]
```

Destructor.

**9.18.4 Member Function Documentation****9.18.4.1 Comment()**

```
const char* FCFSupport::CarType::Comment ( ) const [inline]
```

Return the car type's commentary.

References comment.

**9.18.4.2 Group()**

```
char FCFSupport::CarType::Group ( ) const [inline]
```

Return the car type's group code.

References group.

**9.18.4.3 operator=()**

```
CarType& FCFSupport::CarType::operator= (
    CarType & other ) [inline]
```

Assignment operator.

Copy a car type from another instance.



#### Parameters

<i>other</i>	The other instance.
--------------	---------------------

References comment, group, and type.

#### 9.18.4.4 Type()

```
const char* FCFSupport::CarType::Type ( ) const [inline]
```

Return the car type name.

References type.

### 9.18.5 Friends And Related Function Documentation

#### 9.18.5.1 System

```
friend class System [friend]
```

The [System](#) class is a friend.

### 9.18.6 Member Data Documentation

#### 9.18.6.1 comment

```
string FCFSupport::CarType::comment [private]
```

The commentary string.

Referenced by CarType(), Comment(), and operator=().

### 9.18.6.2 group

```
char FCFSupport::CarType::group [private]
```

The group code.

Referenced by FCFSupport::CarGroup::CarGroup(), CarType(), Group(), FCFSupport::CarGroup::Group(), operator=(), and FCFSupport::CarGroup::operator=().

### 9.18.6.3 type

```
string FCFSupport::CarType::type [private]
```

The type name.

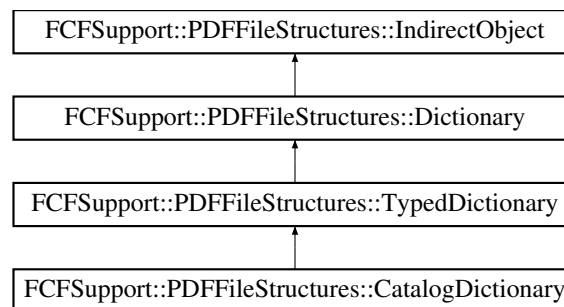
Referenced by CarType(), operator=(), and Type().

## 9.19 FCFSupport::PDFFileStructures::CatalogDictionary Class Reference

Master catalog of the PDF file.

```
#include <PDFPrinterSupport.h>
```

Inheritance diagram for FCFSupport::PDFFileStructures::CatalogDictionary:



### Public Member Functions

- [CatalogDictionary](#) (unsigned long int objNum=0L, unsigned short int genNum=0, CrossReferenceTable \*tab=N←ULL)  
*Constructor.*
- [~CatalogDictionary](#) ()  
*Destructor.*
- void [AddPage](#) ([Page](#) \*thepage)  
*Add a page.*
- void [AddPageTree](#) ([PageTree](#) \*thepagetree)  
*Add a tree of pages.*
- void [AddPageLabelTree](#) ([PageLabelTree](#) \*node)  
*Add a page label tree node.*
- void [AddPageLabelDictionary](#) (int number, [PageLabelDictionary](#) \*pId)  
*Add a page label dictionary.*

## Protected Member Functions

- virtual ostream & [WriteDictionaryElements](#) (ostream &stream) const  
*Write an object directly.*

## Private Attributes

- [PageTree](#) \* [pages](#)  
*Pages.*
- [PageLabelTree](#) \* [labels](#)  
*Page labels.*

### 9.19.1 Detailed Description

Master catalog of the PDF file.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.19.2 Constructor & Destructor Documentation

#### 9.19.2.1 CatalogDictionary()

```
FCFSupport::PDFFileStructures::CatalogDictionary::CatalogDictionary (
    unsigned long int objNum = 0L,
    unsigned short int genNum = 0,
    CrossReferenceTable * tab = NULL ) [inline]
```

Constructor.

#### Parameters

<i>objNum</i>	The next object number.
<i>genNum</i>	The generation number.
<i>tab</i>	The cross reference table we are in.

#### 9.19.2.2 ~CatalogDictionary()

```
FCFSupport::PDFFileStructures::CatalogDictionary::~~CatalogDictionary ( ) [inline]
```

Destructor.

### 9.19.3 Member Function Documentation

#### 9.19.3.1 AddPage()

```
void FCFSupport::PDFFileStructures::CatalogDictionary::AddPage (  
    Page * thepage ) [inline]
```

Add a page.

##### Parameters

<i>thepage</i>	The page to add.
----------------	------------------

#### 9.19.3.2 AddPageLabelDictionary()

```
void FCFSupport::PDFFileStructures::CatalogDictionary::AddPageLabelDictionary (  
    int number,  
    PageLabelDictionary * pld ) [inline]
```

Add a page label dictionary.

##### Parameters

<i>number</i>	The page label dictionary start page number.
<i>pld</i>	Page label dictionary pointer.

References lcc::stream.

#### 9.19.3.3 AddPageLabelTree()

```
void FCFSupport::PDFFileStructures::CatalogDictionary::AddPageLabelTree (  
    PageLabelTree * node ) [inline]
```

Add a page label tree node.

## Parameters

<i>node</i>	The page label tree node.
-------------	---------------------------

References [FCFSupport::PDFFileStructures::PageLabelTree::AddPageLabelTree\(\)](#).

## 9.19.3.4 AddPageTree()

```
void FCFSupport::PDFFileStructures::CatalogDictionary::AddPageTree (
    PageTree * thepagetree ) [inline]
```

Add a tree of pages.

## Parameters

<i>thepagetree</i>	The page tree to add.
--------------------	-----------------------

References [FCFSupport::PDFFileStructures::PageTree::AddPageTree\(\)](#).

## 9.19.3.5 WriteDictionaryElements()

```
virtual ostream& FCFSupport::PDFFileStructures::CatalogDictionary::WriteDictionaryElements (
    ostream & stream ) const [protected], [virtual]
```

Write an object directly.

## Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

Reimplemented from [FCFSupport::PDFFileStructures::TypedDictionary](#).

## 9.19.4 Member Data Documentation

## 9.19.4.1 labels

```
PageLabelTree* FCFSupport::PDFFileStructures::CatalogDictionary::labels [private]
```

[Page](#) labels.

### 9.19.4.2 pages

`PageTree*` `FCFSupport::PDFFileStructures::CatalogDictionary::pages` [private]

Pages.

## 9.20 cmri::CMri Class Reference

Main C/MRI interface class.

### Public Member Functions

- `CMri` (name, port,...)  
*Constructor.*
- `~CMri` ()  
*The destructor restores the serial port's state and closes it.*
- `Inputs` (ni, ua=0)  
*The `Inputs()` function polls the interface and collects the input port values returned by the serial card.*
- `Outputs` (ports, ua=0)  
*The `Outputs()` function sends bytes to the output ports managed by the specified card.*
- `InitBoard` (CT, ni, no, ns, ua, card, dl)  
*The `InitBoard()` function initializes a given USIC, SUSIC, or SMINI card.*

### Private Member Functions

- `_transmit` (ua, mt, ob)  
*Data transmitter.*
- `_readevent` ()  
*Read event method.*
- `_readbyte` (thebytevar)  
*Read a single byte from the serial interface.*

### Private Attributes

- `ttyfd`  
*Terminal file descriptor.*
- `_timeout`  
*Timeout flag.*

## Static Private Attributes

- static [CardType\\_Byte](#)  
*Array of CardType code bytes.*
- static [STX](#)  
*Start of Text.*
- static [ETX](#)  
*End of text.*
- static [DLE](#)  
*Data Link Escape.*
- static [AddressCode](#)  
*Address code.*
- static [Init](#)  
*Initialize message.*
- static [Transmit](#)  
*Transmit message.*
- static [Poll](#)  
*Poll message.*
- static [Read](#)  
*Read message.*

### 9.20.1 Detailed Description

Main C/MRI interface class.

This class implements the interface logic for all of the boards on a given serial bus, attached to a given serial (COM) port. This class effectively implements in Tcl what the QBasic serial I/O subroutines implemented by Bruce Chubb implement under MS-Windows.

The constructor opens the serial port and does low-level serial I/O setup (BAUD rate, etc.). This is the first part of the INIT subroutine.

The [InitBoard\(\)](#) member function initializes a selected board (the second part of the INIT subroutine) and the [Inputs\(\)](#) and [Output\(\)](#) member functions correspond to the INPUTS and OUTPUTS subroutines.

The private members, [\\_transmit\(\)](#) and [\\_readbyte\(\)](#) correspond to the TXPACK and RXBYTE subroutines.

#### Parameters

<i>port</i>	Name of the serial port connected to the Chubb RS485 bus.
...	Options: <ul style="list-style-type: none"> <li>• -baud Data rate, readonly, defaults to 9600, can be one of 9600, 19200, 28800, 57600, or 115200.</li> <li>• -maxtries The maximum number of tries when reading the bus. It is readonly and defaults to 10000. Must be an integer between 1000 and 100000.</li> </ul>

**Author**

Robert Heller <heller@deepsoft.com>

**9.20.2 Constructor & Destructor Documentation****9.20.2.1 CMri()**

```
cmri::CMri::CMri (
    name ,
    port ,
    ... )
```

Constructor.

**Parameters**

<i>port</i>	Name of the serial port connected to the Chubb RS485 bus.
...	Options: <ul style="list-style-type: none"><li>• -baud Data rate, readonly, defaults to 9600, can be one of 9600, 19200, 28800, 57600, or 115200.</li><li>• -maxtries The maximum number of tries when reading the bus. It is readonly and defaults to 10000. Must be an integer between 1000 and 100000.</li></ul>

**9.20.2.2 ~CMri()**

```
cmri::CMri::~~CMri ( )
```

The destructor restores the serial port's state and closes it.

**9.20.3 Member Function Documentation**



### 9.20.3.1 \_readbyte()

```
cmri::CMri::_readbyte (
    thebytevar ) [private]
```

Read a single byte from the serial interface.

Used by the [Inputs\(\)](#) function. Returns false on error and true on success.

#### Parameters

<i>thebytevar</i>	A name of a variable to put the byte read. Undefined if there was an error.
-------------------	---

### 9.20.3.2 \_readevent()

```
cmri::CMri::_readevent ( ) [private]
```

Read event method.

### 9.20.3.3 \_transmit()

```
cmri::CMri::_transmit (
    ua ,
    mt ,
    ob ) [private]
```

Data transmitter.

The data is built into a proper message and sent out the serial port to the selected card. Returns false on error and true on success.

#### Parameters

<i>ua</i>	The card address.
<i>mt</i>	The message type.
<i>ob</i>	The data buffer (not used for Poll messages).

### 9.20.3.4 InitBoard()

```
cmri::CMri::InitBoard (
```

```

    CT ,
    ni ,
    no ,
    ns ,
    ua ,
    card ,
    dl )

```

The [InitBoard\(\)](#) function initializes a given USIC, SUSIC, or SMINI card.

#### Parameters

<i>CT</i>	The card type / yellow bi-color LED map. For USIC and SUSIC cards this is the card type map. For the SMINI card this is a 6 element list containing the port pairs for any simulated yellow bi-color LEDs.
-----------	--

The card type map for USIC and SUSIC is a packed array of 2-bit values, packed 4 per element (byte) from low to high. Each 2-bit value is one of 0 (for no card), 1 (for an input card), or 2 (for an output card). The cards must be "packed" with no open slots except at the end of the bus.

For the simulated yellow LEDs (SMINI card) the paired bits must be adjacent red/green bits and cannot span ports.

#### Parameters

<i>ni</i>	The total number of input ports (must be 3 for SMINI).
<i>no</i>	The total number of output ports (must be 6 or SMINI).
<i>ns</i>	The number of yellow bi-color LED signals. Only used for SMINI cards. For USIC and SUSIC cards the Length() member function of the CT parameter is used.
<i>ua</i>	The card address.
<i>card</i>	The card type.
<i>dl</i>	The delay value to use.

#### 9.20.3.5 Inputs()

```

cmri::CMri::Inputs (
    ni ,
    ua = 0 )

```

The [Inputs\(\)](#) function polls the interface and collects the input port values returned by the serial card.

The result is a freshly allocated List object. The calling program should free this memory with delete(). [Inputs\(\)](#) returns a NULL pointer if there was an error.

#### Parameters

<i>ni</i>	The number of input ports to be read. Must equal the number of ports on the specified card.
<i>ua</i>	The card address.

### 9.20.3.6 Outputs()

```
cmri::CMri::Outputs (
    ports ,
    ua = 0 )
```

The [Outputs\(\)](#) function sends bytes to the output ports managed by the specified card.

Since each element is written to one 8-bit output port, each element is presumed to be a integer in the range of 0 to 255.

#### Parameters

<i>ports</i>	The list of port values. Should have as many elements as there are output ports.
<i>ua</i>	The card address.

## 9.20.4 Member Data Documentation

### 9.20.4.1 \_timeout

```
cmri::CMri::_timeout [private]
```

Timeout flag.

### 9.20.4.2 AddressCode

```
cmri::CMri::AddressCode [static], [private]
```

Address code.

### 9.20.4.3 CardType\_Byte

```
cmri::CMri::CardType_Byte [static], [private]
```

Array of CardType code bytes.

#### 9.20.4.4 DLE

```
cmri::CMri::DLE [static], [private]
```

Data Link Escape.

Used to escape special codes.

#### 9.20.4.5 ETX

```
cmri::CMri::ETX [static], [private]
```

End of text.

Used at the end of message blocks.

#### 9.20.4.6 Init

```
cmri::CMri::Init [static], [private]
```

Initialize message.

Initialize a serial interface board.

#### 9.20.4.7 Poll

```
cmri::CMri::Poll [static], [private]
```

Poll message.

Request the board to read its input ports.

#### 9.20.4.8 Read

```
cmri::CMri::Read [static], [private]
```

Read message.

Generated by a board in response to a Poll message.

#### 9.20.4.9 STX

```
cmri::CMri::STX [static], [private]
```

Start of Text.

Used at the start of message blocks.

## 9.20.4.10 Transmit

```
cmri::Cmri::Transmit [static], [private]
```

Transmit message.

Send data to output ports.

## 9.20.4.11 ttyfd

```
cmri::Cmri::ttyfd [private]
```

Terminal file descriptor.

## 9.21 CmriSupport::CmriNode Class Reference

CMR/I node type.

### Public Member Functions

- [CmriNode](#) (name,...)  
*Constructor – initialize a board.*
- [inputs](#) ()  
*Method to fetch input port values.*
- [outputs](#) (portvector="")  
*Method to set output ports.*
- [setport](#) (portnum, byte)  
*Set and send one byte to a port (rewrites all ports).*
- [setbitfield](#) (portnum, mask, bits)  
*Set and send a bitfield to a port (rewrites all ports).*

### Static Public Member Functions

- static [validate](#) (object)  
*Type validating code Raises an error if object is not either the empty string or a C4TSMINI\_Block type.*
- static [openport](#) (port="/dev/ttyS0", baud=9600, maxtries=10000)  
*Open the CMR/I port.*
- static [closeport](#) ()  
*Close the CMR/I port.*
- static [portopenp](#) ()  
*Return port status.*

## Public Attributes

- [outputbuffer](#)  
*Output buffer.*

## Private Member Functions

- [\\_ValidateType](#) (option, value)  
*Method to validate the card type.*
- [\\_ValidateAddress](#) (option, value)  
*Method to validate a card address.*
- [\\_ValidateListOfBytes](#) (option, value)  
*Method to validate a list of bytes.*
- [\\_ValidateSixElementListOfBytes](#) (option, value)  
*Method to validate a six element list of bytes.*
- [\\_ValidateByte](#) (option, value)  
*Method to validate a byte value.*
- [\\_ValidateWord](#) (option, value)  
*Method to validate a word (16-bit) value.*

## Static Private Attributes

- static [\\_TypeCodes](#)  
*Node type codes.*

### 9.21.1 Detailed Description

CMR/I node type.

This Snit type defines CMR/I nodes (SUSIC, USIC, or SMINI boards) on a CMR/I network. All options are readonly.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.21.2 Constructor & Destructor Documentation

#### 9.21.2.1 CmriNode()

```
CmriSupport::CmriNode::CmriNode (
    name ,
    ... )
```

Constructor – initialize a board.

## Parameters

<i>name</i>	Name of the node.
...	Options: <ul style="list-style-type: none"> <li>• <code>-type</code> The type of node, one of SUSIC, USIC, or SMINI. No default value.</li> <li>• <code>-address</code> The address of the node. Default is 0.</li> <li>• <code>-cardmap</code> The card type map. Only used with SUSIC and USIC. Default is {}.</li> <li>• <code>-yellowmap</code> The yellow bi-color LED map. Only used with the SMINI card type. Default is {0 0 0 0 0 0}.</li> <li>• <code>-numberofyellow</code> The number of yellow bi-color LED signals. Only for SMINI cards. Default is 0.</li> <li>• <code>-inputports</code> The number of 8-bit input ports. Default 0 (3 for SMINI cards).</li> <li>• <code>-outputports</code> The number of 8-bit output ports. Default 0 (6 for SMINI cards).</li> <li>• <code>-delay</code> The delay value to use. Only meaningful for older (USIC) cards. Default is 0.</li> </ul>

## 9.21.3 Member Function Documentation

## 9.21.3.1 \_ValidateAddress()

```
CmriSupport::CmriNode::_ValidateAddress (
    option ,
    value ) [private]
```

Method to validate a card address.

## Parameters

<i>option</i>	The option to validate.
<i>value</i>	The value to validate.

## 9.21.3.2 \_ValidateByte()

```
CmriSupport::CmriNode::_ValidateByte (
    option ,
    value ) [private]
```

Method to validate a byte value.



## Parameters

<i>option</i>	The option to validate.
<i>value</i>	The value to validate.

## 9.21.3.3 \_ValidateListOfBytes()

```
CmriSupport::CmriNode::_ValidateListOfBytes (  
    option ,  
    value ) [private]
```

Method to validate a list of bytes.

## Parameters

<i>option</i>	The option to validate.
<i>value</i>	The value to validate.

## 9.21.3.4 \_ValidateSixElementListOfBytes()

```
CmriSupport::CmriNode::_ValidateSixElementListOfBytes (  
    option ,  
    value ) [private]
```

Method to validate a six element list of bytes.

## Parameters

<i>option</i>	The option to validate.
<i>value</i>	The value to validate.

## 9.21.3.5 \_ValidateType()

```
CmriSupport::CmriNode::_ValidateType (  
    option ,  
    value ) [private]
```

Method to validate the card type.

**Parameters**

<i>option</i>	The option to validate.
<i>value</i>	The value to validate.

**9.21.3.6 \_ValidateWord()**

```
CmriSupport::CmriNode::_ValidateWord (
    option ,
    value ) [private]
```

Method to validate a word (16-bit) value.

**Parameters**

<i>option</i>	The option to validate.
<i>value</i>	The value to validate.

**9.21.3.7 closeport()**

```
static CmriSupport::CmriNode::closeport ( ) [static]
```

Close the CMR/I port.

This type method closes the CMR/I port.

**9.21.3.8 inputs()**

```
CmriSupport::CmriNode::inputs ( )
```

Method to fetch input port values.

**9.21.3.9 openport()**

```
static CmriSupport::CmriNode::openport (
    port = "/dev/ttyS0",
    baud = 9600,
    maxtries = 10000 ) [static]
```

Open the CMR/I port.

This type method opens the CMR/I port.

## Parameters

<i>port</i>	The serial port connected to the CMR/I network.
<i>baud</i>	The BAUD rate to be used.
<i>maxtries</i>	The maximum number of retries.

## 9.21.3.10 outputs()

```
CmriSupport::CmriNode::outputs (
    portvector = "" )
```

Method to set output ports.

## Parameters

<i>portvector</i>	Vector of output ports.
-------------------	-------------------------

## 9.21.3.11 portopenp()

```
static CmriSupport::CmriNode::portopenp ( ) [static]
```

Return port status.

## 9.21.3.12 setbitfield()

```
CmriSupport::CmriNode::setbitfield (
    portnum ,
    mask ,
    bits )
```

Set and send a bitfield to a port (rewrites all ports).

## Parameters

<i>portnum</i>	Number of the output port.
<i>mask</i>	Bit mask.
<i>bits</i>	Bits (must already shifted into position!).

#### 9.21.3.13 setport()

```
CmriSupport::CmriNode::setport (
    portnum ,
    byte )
```

Set and send one byte to a port (rewrites all ports).

##### Parameters

<i>portnum</i>	Number of the output port.
<i>byte</i>	Value to write.

#### 9.21.3.14 validate()

```
static CmriSupport::CmriNode::validate (
    object ) [static]
```

Type validating code Raises an error if object is not either the empty string or a C4TSMINI\_Block type.

### 9.21.4 Member Data Documentation

#### 9.21.4.1 \_TypeCodes

```
CmriSupport::CmriNode::_TypeCodes [static], [private]
```

Node type codes.

#### 9.21.4.2 outputbuffer

```
CmriSupport::CmriNode::outputbuffer
```

Output buffer.

## 9.22 CTCPanel::CodeButton Class Reference

Code button object type.

### Public Member Functions

- [CodeButton](#) (name, \_ctcpanel, \_canvas,...)  
*Construct a Code Button object.*
- [~CodeButton](#) ()  
*Clean up all data objects and free up all resources.*
- [getv](#) ()  
*Method to get our value (none).*
- [setv](#) (state)  
*Method to set our value (none).*
- [geti](#) (ind)  
*Method to get an indicator state (none).*
- [seti](#) (ind, value)  
*Method to set an indicator state (none).*
- [invoke](#) ()  
*Method to invoke the code button.*

### Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*

### 9.22.1 Detailed Description

Code button object type.

These are on the control panel and represent buttons that enact the settings of the SWPlates and SIGPlates for a given control point.

#### Parameters

<a href="#">_ctcpanel</a>	The <a href="#">CTCPanel</a> megawidget.
<a href="#">_canvas</a>	The control panel canvas to draw the switch plate on.
...	Options: <ul style="list-style-type: none"> <li>• -x The x coordinate of the object (readonly, default 0).</li> <li>• -y The y coordinate of the object (readonly, default 0).</li> <li>• -controlpoint The name of the control point this switch is part of (readonly, default CP1).</li> <li>• -command The Tcl script to run when the code button is invoked.</li> </ul>

Defined coords terminals: none. Defined values (states): none. Defined indicators: none.

#### Author

Robert Heller <heller@deepsoft.com>

## 9.22.2 Constructor & Destructor Documentation

### 9.22.2.1 CodeButton()

```
CTCPanel::CodeButton::CodeButton (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a Code Button object.

#### Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The control panel canvas to draw the Code Button on.
<code>...</code>	Option list.

### 9.22.2.2 ~CodeButton()

```
CTCPanel::CodeButton::~~CodeButton ( )
```

Clean up all data objects and free up all resources.

## 9.22.3 Member Function Documentation

### 9.22.3.1 geti()

```
CTCPanel::CodeButton::geti (
    ind )
```

Method to get an indicator state (none).

### 9.22.3.2 getv()

```
CTCPanel::CodeButton::getv ( )
```

Method to get our value (none).

### 9.22.3.3 invoke()

```
CTCPanel::CodeButton::invoke ( )
```

Method to invoke the code button.

The command script is executed.

### 9.22.3.4 seti()

```
CTCPanel::CodeButton::seti (
    ind ,
    value )
```

Method to set an indicator state (none).

### 9.22.3.5 setv()

```
CTCPanel::CodeButton::setv (
    state )
```

Method to set our value (none).

## 9.22.4 Member Data Documentation

### 9.22.4.1 canvas

```
CTCPanel::CodeButton::canvas [private]
```

The canvas component (parent widget component).

#### 9.22.4.2 ctcp panel

CTCPanel::CodeButton::ctcp panel [private]

The CTC Panel component (parent widget).

### 9.23 xpressnet::CommandStationResponse Class Reference

General response class.

#### Public Member Functions

- [TimeStamp](#) ()  
*Return the time stamp of the response.*
- [ResponseType](#) ()  
*Return the response type.*
- [CommandStationResponse](#) (name,...)  
*Constructor.*

#### Private Attributes

- [\\_time\\_stamp](#)  
*Holds the time stamp of the response.*

#### 9.23.1 Detailed Description

General response class.

All responses are delegated from this class, via a component element.

##### Parameters

<code>-response type</code>	This readonly option contains the response type and determines the type of object installed in the actual response component.
-----------------------------	---

Additional parameters are passed to the actual response constructors.

#### Author

Robert Heller <heller@deepsoft.com>



## 9.23.2 Constructor & Destructor Documentation

### 9.23.2.1 CommandStationResponse()

```
xpressnet::CommandStationResponse::CommandStationResponse (
    name ,
    ... )
```

Constructor.

Construct a response object. The actual response is installed as a component of this object.

Parameters

<i>-responsetype</i>	This readonly option contains the response type and determines the type of object installed in the actual response component.
----------------------	---

Additional parameters are passed to the actual response constructors.

## 9.23.3 Member Function Documentation

### 9.23.3.1 ResponseType()

```
xpressnet::CommandStationResponse::ResponseType ( )
```

Return the response type.

### 9.23.3.2 TimeStamp()

```
xpressnet::CommandStationResponse::TimeStamp ( )
```

Return the time stamp of the response.

## 9.23.4 Member Data Documentation

#### 9.23.4.1 `_time_stamp`

`xpressnet::CommandStationResponse::_time_stamp` [private]

Holds the time stamp of the response.

### 9.24 `xpressnet::CommandStationStatus` Class Reference

Command station status.

#### Public Member Functions

- [`CommandStationStatus`](#) (name, statusbyte)  
*Constructor.*
- [`EmergencyOff`](#) ()  
*Return emergency off flag.*
- [`EmergencyStop`](#) ()  
*Return emergency stop flag.*
- [`StartMode`](#) ()  
*Return start mode.*
- [`ServiceMode`](#) ()  
*Return service mode.*
- [`PoweringUp`](#) ()  
*Return powering up flag.*
- [`RAMCheckError`](#) ()  
*Return RAM check error flag.*

#### Private Attributes

- [`\_emergency\_off`](#)  
*Emergency off flag.*
- [`\_emergency\_stop`](#)  
*Emergency stop flag.*
- [`\_start\_mode`](#)  
*Start mode.*
- [`\_service\_mode`](#)  
*Service mode flag.*
- [`\_poweringup`](#)  
*Powering up flag.*
- [`\_RAM\_check\_error`](#)  
*RAM check error flag.*

### 9.24.1 Detailed Description

Command station status.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.24.2 Constructor & Destructor Documentation

#### 9.24.2.1 CommandStationStatus()

```
xpressnet::CommandStationStatus::CommandStationStatus (
    name ,
    statusbyte )
```

Constructor.

#### Parameters

<i>statusbyte</i>	Status byte.
-------------------	--------------

### 9.24.3 Member Function Documentation

#### 9.24.3.1 EmergencyOff()

```
xpressnet::CommandStationStatus::EmergencyOff ( )
```

Return emergency off flag.

#### 9.24.3.2 EmergencyStop()

```
xpressnet::CommandStationStatus::EmergencyStop ( )
```

Return emergency stop flag.

#### 9.24.3.3 PoweringUp()

```
xpressnet::CommandStationStatus::PoweringUp ( )
```

Return powering up flag.

#### 9.24.3.4 RAMCheckError()

```
xpressnet::CommandStationStatus::RAMCheckError ( )
```

Return RAM check error flag.

#### 9.24.3.5 ServiceMode()

```
xpressnet::CommandStationStatus::ServiceMode ( )
```

Return service mode.

#### 9.24.3.6 StartMode()

```
xpressnet::CommandStationStatus::StartMode ( )
```

Return start mode.

### 9.24.4 Member Data Documentation

#### 9.24.4.1 \_emergency\_off

```
xpressnet::CommandStationStatus::_emergency_off [private]
```

Emergency off flag.

#### 9.24.4.2 \_emergency\_stop

xpressnet::CommandStationStatus::\_emergency\_stop [private]

Emergency stop flag.

#### 9.24.4.3 \_poweringup

xpressnet::CommandStationStatus::\_poweringup [private]

Powering up flag.

#### 9.24.4.4 \_RAM\_check\_error

xpressnet::CommandStationStatus::\_RAM\_check\_error [private]

RAM check error flag.

#### 9.24.4.5 \_service\_mode

xpressnet::CommandStationStatus::\_service\_mode [private]

Service mode flag.

#### 9.24.4.6 \_start\_mode

xpressnet::CommandStationStatus::\_start\_mode [private]

Start mode.

## 9.25 Parsers::TrackGraph::CompressedEdgeValues Struct Reference

Compressed graph edge values.

## Public Member Functions

- [CompressedEdgeValues](#) (float \_length=0.0)  
*Default constructor.*

## Public Attributes

- float [length](#)  
*Track length from opposite edge.*

### 9.25.1 Detailed Description

Compressed graph edge values.

### 9.25.2 Constructor & Destructor Documentation

#### 9.25.2.1 CompressedEdgeValues()

```
Parsers::TrackGraph::CompressedEdgeValues::CompressedEdgeValues (
    float _length = 0.0 ) [inline]
```

Default constructor.

### 9.25.3 Member Data Documentation

#### 9.25.3.1 length

```
float Parsers::TrackGraph::CompressedEdgeValues::length
```

Track length from opposite edge.

#### Author

Robert Heller <heller@deepsoft.com>

## 9.26 Parsers::TrackGraph::CompressedNodeValues Struct Reference

Compressed graph node values.

## Public Member Functions

- `std::list< int >::size_type` [FindSegmentIndex](#) (int segment) const  
*Return the segment index for a given segment.*
- [CompressedNodeValues](#) (int \_id=-1)  
*Default constructor.*

## Public Attributes

- int [id](#)  
*Node number.*
- [Node rawnode](#)  
*Uncompressed head node for this compressed node.*
- [Point position](#)  
*Node's graphical position.*
- `std::list< int >` [segments](#)  
*List of uncompressed node ids.*

### 9.26.1 Detailed Description

Compressed graph node values.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.26.2 Constructor & Destructor Documentation

#### 9.26.2.1 CompressedNodeValues()

```
Parsers::TrackGraph::CompressedNodeValues::CompressedNodeValues (  
    int _id = -1 ) [inline]
```

Default constructor.

References `Parsers::TrackGraph::Point::x`, and `Parsers::TrackGraph::Point::y`.

### 9.26.3 Member Function Documentation

#### 9.26.3.1 FindSegmentIndex()

```
std::list<int>::size_type Parsers::TrackGraph::CompressedNodeValues::FindSegmentIndex (
    int segment ) const [inline]
```

Return the segment index for a given segment.

### 9.26.4 Member Data Documentation

#### 9.26.4.1 id

```
int Parsers::TrackGraph::CompressedNodeValues::id
```

Node number.

#### 9.26.4.2 position

```
Point Parsers::TrackGraph::CompressedNodeValues::position
```

Node's graphical position.

#### 9.26.4.3 rawnode

```
Node Parsers::TrackGraph::CompressedNodeValues::rawnode
```

Uncompressed head node for this compressed node.

#### 9.26.4.4 segments

```
std::list<int> Parsers::TrackGraph::CompressedNodeValues::segments
```

List of uncompressed node ids.



## 9.27 Icc::ConfigMemory Class Reference

Configure memory.

### Public Member Functions

- [ConfigMemory](#) (name,...)  
*Construct a memory config dialog.*
- [\\_Close](#) ()  
*Close and destroy the dialog box.*
- [\\_Read](#) ()  
*Bound to the `Read` button.*
- [\\_Write](#) ()  
*Bound to the `Write` button.*
- [\\_getAddressRange](#) (thespace)  
*Get the address range of the specified space.*
- [\\_Dump](#) ()  
*Bound to the `Dump` button.*
- [\\_dumpAsText](#) (thespace, startaddress, endaddress)  
*Dump a space as text (typically the CDI).*
- [\\_dumpAsHex](#) (thespace, startaddress, endaddress)  
*Dump a space as hex (typically the configuration memory).*
- [\\_Restore](#) ()  
*Bound to the `Restore` button.*

### Private Member Functions

- [\\_datagramhandler](#) (command, sourcenid,...)  
*Datagram message handler.*
- [\\_messagehandler](#) (message)  
*Message handler – handle incoming messages.*
- [\\_readmemory](#) (\_space, \_address, length, status\_var)  
*Method to read a block of configuration memory.*
- [\\_writememory](#) (\_space, \_address, databuffer)  
*Write a block of data to configuration memory.*
- [putdebug](#) (message)  
*Print message using debug output, if any.*

## Private Attributes

- [readlist](#)  
*Read list.*
- [writelist](#)  
*Write list.*
- [count](#)  
*Byte count.*
- [address](#)  
*Start address.*
- [space](#)  
*Space select.*
- [\\_ioComplete](#)  
*I/O Completion Flag.*
- [olddatagramhandler](#)  
*Old datagram handler.*
- [oldgeneralmessagehandler](#)  
*Old general message handler.*
- [datagrambuffer](#)  
*Datagram message buffer.*
- [\\_datagramrejecterror](#)  
*Last datagram rejection error.*
- [writeReplyCheck](#)  
*Flag to check for a write reply.*

## Static Private Attributes

- static [\\_spaces](#)  
*Space values.*

### 9.27.1 Detailed Description

Configure memory.

Create a dialog box that reads and writes the configuration memory of an OpenLCB node.

Options:

- -destnid Node ID to send to.
- -transport LCC Transport object.
- -debugprint A function to handle debug output.

## 9.27.2 Constructor & Destructor Documentation

### 9.27.2.1 ConfigMemory()

```
lcc::ConfigMemory::ConfigMemory (
    name ,
    ... )
```

Construct a memory config dialog.

#### Parameters

<i>name</i>	Pathname of the widget.
...	Options: <ul style="list-style-type: none"><li>• -destnid Node ID to send to.</li><li>• -transport LCC Transport object.</li></ul>

## 9.27.3 Member Function Documentation

### 9.27.3.1 \_Close()

```
lcc::ConfigMemory::_Close ( )
```

Close and destroy the dialog box.

### 9.27.3.2 \_datagramhandler()

```
lcc::ConfigMemory::_datagramhandler (
    command ,
    sourcenid ,
    ... ) [private]
```

Datagram message handler.

This method is called when a datagram type message arrives.

## Parameters

<i>command</i>	One of datagramreceivedok, datagramrejected, or datagramcontent.
<i>sourcenid</i>	The Node ID of the node sending the datagram.
...	The data buffer, if any. <ul style="list-style-type: none"> <li>-debugprint A function to handle debug output.</li> </ul>

## 9.27.3.3 \_Dump()

```
lcc::ConfigMemory::_Dump ( )
```

Bound to the Dump button.

Dump the configuration memory to a file. Either as text (if space is CDI) or Hex (if space is NOT CDI).

## 9.27.3.4 \_dumpAsHex()

```
lcc::ConfigMemory::_dumpAsHex (
    thespace ,
    startaddress ,
    endaddress )
```

Dump a space as hex (typically the configuration memory).

Dump a device's memory as a hex file. This is typically the device's configuration memory.

## Parameters

<i>thespace</i>	The space.
<i>startaddress</i>	The start address
<i>endaddress</i>	The end address

## 9.27.3.5 \_dumpAsText()

```
lcc::ConfigMemory::_dumpAsText (
    thespace ,
    startaddress ,
    endaddress )
```

Dump a space as text (typically the CDI).

Dump a device's memory as a text file. This is typically the device's CDI.

## Parameters

<i>thespace</i>	The space.
<i>startaddress</i>	The start address
<i>endaddress</i>	The end address

9.27.3.6 `_getAddressRange()`

```
lcc::ConfigMemory::_getAddressRange (
    thespace )
```

Get the address range of the specified space.

This performs a Get Address Space Information Command and then returns the address range info.

## Parameters

<i>thespace</i>	The space.
-----------------	------------

9.27.3.7 `_messagehandler()`

```
lcc::ConfigMemory::_messagehandler (
    message ) [private]
```

Message handler – handle incoming messages.

Certain messages are processed:

Initialization Complete Messages – This is a possible response to freeze, unfreeze, reset, or reinitialize commands.

9.27.3.8 `_Read()`

```
lcc::ConfigMemory::_Read ( )
```

Bound to the `Read` button.

Read a block of memory and display the results.

### 9.27.3.9 \_readmemory()

```
lcc::ConfigMemory::_readmemory (
    _space ,
    _address ,
    length ,
    status_var ) [private]
```

Method to read a block of configuration memory.

Read a block of memory, return the data bytes. The variable named by the `status_var` is side effected with the status code.

#### Parameters

<code>_space</code>	The memory space to read from.
<code>_address</code>	The address to start reading from.
<code>length</code>	The number of bytes to read.
<code>status_var</code>	The name of a status variable.

#### Returns

The data block read.

### 9.27.3.10 \_Restore()

```
lcc::ConfigMemory::_Restore ( )
```

Bound to the `Restore` button.

Reload configuration memory from a hex dump file.

### 9.27.3.11 \_Write()

```
lcc::ConfigMemory::_Write ( )
```

Bound to the `Write` button.

Write a block of memory.

### 9.27.3.12 \_writememory()

```
lcc::ConfigMemory::_writememory (
    _space ,
    _address ,
    databuffer ) [private]
```

Write a block of data to configuration memory.

This method writes a block of memory to configuration memory of an OpenLCB node.

**Parameters**

<i>_space</i>	The memory space to write to.
<i>_address</i>	The address to start writing to.
<i>databuffer</i>	The list of bytes to write.

**Returns**

The result status: 0 if successful, otherwise an error code.

**9.27.3.13 putdebug()**

```
lcc::ConfigMemory::putdebug (
    message ) [private]
```

Print message using debug output, if any.

**Parameters**

<i>message</i>	The message to print.
----------------	-----------------------

**9.27.4 Member Data Documentation****9.27.4.1 \_datagramrejecterror**

```
lcc::ConfigMemory::_datagramrejecterror [private]
```

Last datagram rejection error.

**9.27.4.2 \_ioComplete**

```
lcc::ConfigMemory::_ioComplete [private]
```

I/O Completion Flag.

#### 9.27.4.3 `_spaces`

```
lcc::ConfigMemory::_spaces [static], [private]
```

Space values.

#### 9.27.4.4 `address`

```
lcc::ConfigMemory::address [private]
```

Start address.

#### 9.27.4.5 `count`

```
lcc::ConfigMemory::count [private]
```

Byte count.

#### 9.27.4.6 `datagrambuffer`

```
lcc::ConfigMemory::datagrambuffer [private]
```

Datagram message buffer.

#### 9.27.4.7 `olddatagramhandler`

```
lcc::ConfigMemory::olddatagramhandler [private]
```

Old datagram handler.

#### 9.27.4.8 `oldgeneralmessagehandler`

```
lcc::ConfigMemory::oldgeneralmessagehandler [private]
```

Old general message handler.



#### 9.27.4.9 readlist

```
lcc::ConfigMemory::readlist [private]
```

Read list.

#### 9.27.4.10 space

```
lcc::ConfigMemory::space [private]
```

Space select.

#### 9.27.4.11 writelist

```
lcc::ConfigMemory::writelist [private]
```

Write list.

#### 9.27.4.12 writeReplyCheck

```
lcc::ConfigMemory::writeReplyCheck [private]
```

Flag to check for a write reply.

## 9.28 lcc::ConfigOptions Class Reference

Display memory config options.

### Public Member Functions

- [ConfigOptions](#) (name,...)  
*Construct a Config Options dialog.*
- [\\_Close](#) ()  
*Close and destroy the dialog box.*

### Private Member Functions

- `putdebug` (message)  
*Print message using debug output, if any.*

### Private Attributes

- `nodeid`  
*Node ID.*
- `available`  
*Available bits.*
- `writelengths`  
*Write lengths.*
- `highest`  
*Highest memory space.*
- `lowest`  
*Lowest memory space.*
- `name`  
*Name string.*

## 9.28.1 Detailed Description

Display memory config options.

Options

- `-nid` Node ID.
- `-available` Available bitmask.
- `-writelengths` Write length bitmask.
- `-highest` Highest memory space.
- `-lowest` Lowest memory space.
- `-name` Name string.
- `-debugprint` A function to handle debug output.

## 9.28.2 Constructor & Destructor Documentation

### 9.28.2.1 ConfigOptions()

```
lcc::ConfigOptions::ConfigOptions (
    name ,
    ... )
```

Construct a Config Options dialog.

## Parameters

<i>name</i>	The widget pathname.
...	Options: <ul style="list-style-type: none"><li>• -nid Node ID.</li><li>• -available Available bitmask.</li><li>• -writelengths Write length bitmask.</li><li>• -highest Highest memory space.</li><li>• -lowest Lowest memory space.</li><li>• -name Name string.</li><li>• -debugprint A function to handle debug output.</li></ul>

## 9.28.3 Member Function Documentation

## 9.28.3.1 \_Close()

```
lcc::ConfigOptions::_Close ( )
```

Close and destroy the dialog box.

## 9.28.3.2 putdebug()

```
lcc::ConfigOptions::putdebug (
    message ) [private]
```

Print message using debug output, if any.

## Parameters

<i>message</i>	The message to print.
----------------	-----------------------

## 9.28.4 Member Data Documentation

### 9.28.4.1 available

`lcc::ConfigOptions::available` [private]

Available bits.

### 9.28.4.2 highest

`lcc::ConfigOptions::highest` [private]

Highest memory space.

### 9.28.4.3 lowest

`lcc::ConfigOptions::lowest` [private]

Lowest memory space.

### 9.28.4.4 name

`lcc::ConfigOptions::name` [private]

Name string.

### 9.28.4.5 nodeid

`lcc::ConfigOptions::nodeid` [private]

Node ID.

## 9.28.4.6 writelengths

```
lcc::ConfigOptions::writelengths [private]
```

Write lengths.

## 9.29 lcc::ConfigurationEditor Class Reference

Generate OpenLCB Memory Configuration Window.

### Public Member Functions

- [ConfigurationEditor](#) (name,...)  
*Constructor: create the configuration editor.*

### Private Member Functions

- [putdebug](#) (message)  
*Print message using debug output, if any.*
- [\\_processXMLnode](#) (n, frame, space, address\_var, prefix="")  
*Process one node in the XML tree.*
- [\\_printexport](#) (node, frame, name)  
*Print or export a segment or group.*
- [\\_printexport\\_pdf](#) (node, frame, name, outfile)  
*Export a segment or group to a printable PDF file.*
- [\\_printexport\\_xml](#) (node, frame, name, outfile)  
*Export a segment or group to an XML file.*
- [\\_printexport\\_csv](#) (node, frame, name, outfile)  
*Export a segment or group to a CSV file (can be imported into Excel).*
- [\\_printexport\\_txt](#) (node, frame, name, outfile)  
*Export a segment or group to a text file.*
- [\\_close](#) ()  
*Close the window.*
- [\\_datagramhandler](#) (command, sourcenid,...)  
*Datagram handler.*
- [\\_readmemory](#) (space, address, length, status\_var)  
*Read memory from a space.*
- [\\_writememory](#) (space, address, databuffer)  
*Write to configuration memory.*
- [\\_intComboRead](#) (widget, space, address, size)  
*Read an integer value and map it to a ComboBox widget.*
- [\\_intComboWrite](#) (widget, space, address, size, min, max)  
*Write an integer value mapped from a ComboBox widget.*
- [\\_intSpinRead](#) (widget, space, address, size)

- Read an integer value and stash it in a SpinBox widget.*
  - [\\_intSpinWrite](#) (widget, space, address, size, min, max)
- Write an integer value mapped from a SpinBox widget.*
  - [\\_stringComboRead](#) (widget, space, address, size)
- Read a string value and map it to a ComboBox widget.*
  - [\\_stringComboWrite](#) (widget, space, address, size)
- Write a string value mapped from a ComboBox widget.*
  - [\\_stringEntryRead](#) (widget, space, address, size)
- Read a string value and stash it in an Entry widget.*
  - [\\_stringEntryWrite](#) (widget, space, address, size)
- Write a string value from an Entry widget.*
  - [\\_eventidComboRead](#) (widget, space, address, size)
- Read an event id value and map it to a ComboBox widget.*
  - [\\_eventidComboWrite](#) (widget, space, address, size)
- Write an event id value mapped from a ComboBox widget.*
  - [\\_eventidEntryRead](#) (widget, space, address, size)
- Read an event id value and stash it in an Entry widget as an event id string.*
  - [\\_eventidEntryWrite](#) (widget, space, address, size)
- Write an event id value from an Entry widget.*
  - [\\_readall](#) (space)
- Read all parameters stored in a specified space.*

## Static Private Member Functions

- static [\\_printexport\\_pdf\\_frame](#) (n, indent, pdfobj, frame, curyVar, curpageVar, pageheader)*Export a node frame to a PDF file.*
- static [\\_printexport\\_pdf\\_vframe](#) (n, indent, pdfobj, frame, curyVar, curpageVar, pageheader)*Export a node scaler value frame to a PDF file.*
- static [\\_printexport\\_pdf\\_newpage](#) (pdfobj, pageheader, pageno)*Print a new PDF page.*
- static [\\_printexport\\_xml\\_frame](#) (n, frame)*Export a node frame as an XML tree.*
- static [\\_printexport\\_xml\\_vframe](#) (n, frame)*Export a scaler node's value frame as an XML tree.*
- static [\\_printexport\\_csv\\_frame](#) (n, matrix, frame)*Add a node's GUI frame values to a matrix (to be exported as a CSV file).*
- static [\\_printexport\\_csv\\_vframe](#) (n, matrix, frame)*Add a scaler node's GUI value frame values to a matrix (to be exported as a CSV file).*
- static [\\_printexport\\_csv\\_framesAcross](#) (n, tabnb, tabs, matrix)*Add a replicated group to a matrix as a single row.*
- static [\\_printexport\\_csv\\_frameAcross](#) (n, rowVar, frame)*Add a group to a matrix as elements to a single row.*
- static [\\_printexport\\_csv\\_vframeAcross](#) (n, rowVar, frame)*Add a scaler node's value frame to a matrix as elements to a single row.*
- static [\\_printexport\\_txt\\_frame](#) (n, indent, outfp, frame)*Export a segment or group frame to a text file.*
- static [\\_printexport\\_txt\\_vframe](#) (n, indent, outfp, frame)*Export a node scaler value frame to a text file.*

## Private Attributes

- [main](#)  
*Main Frame.*
- [scroll](#)  
*Scrolled Window.*
- [editframe](#)  
*Scrollable Frame.*
- [buttons](#)  
*Button box.*
- [cdi](#)  
*CDI XML Object.*
- [\\_ioComplete](#)  
*I/O Completion Flag.*
- [statusline](#)  
*Status variable.*
- [\\_readall](#)  
*Holds all of the Read buttons for each segment.*
- [\\_segmentnumber](#)  
*Segment number, used to insure unique widget names.*
- [\\_groupnumber](#)  
*Group number, used to insure unique widget names.*
- [\\_intnumber](#)  
*Integer number, used to insure unique widget names.*
- [\\_stringnumber](#)  
*String number, used to insure unique widget names.*
- [\\_eventidnumber](#)  
*Eventid number, used to insure unique widget names.*
- [\\_mkbuttons](#)  
*Flag for Make Sensor / Make Turnout etc.*
- [olddatagramhandler](#)  
*Variable holding the old Datagram handler.*
- [datagrambuffer](#)  
*Datagram buffer.*
- [\\_datagramrejecterror](#)  
*Datagram reject error flag.*
- [writeReplyCheck](#)  
*Datagram write trply check flag.*

## Static Private Attributes

- static [\\_menu](#)  
*Generic menu.*
- static [idheaders](#)  
*Locale versions of the identification headers.*
- static [printexportfiletypes](#)  
*Print and Export file types.*

### 9.29.1 Detailed Description

Generate OpenLCB Memory Configuration Window.

Create a toplevel to configure a node's Memory using that node's (parsed) CDI. This GUI uses tabbed notebook widgets for segments and replicated groups to reduce the amount of scrolling (and because a ginormous scrollable frame dies with a X11 Pixmap allocation error).

#### Parameters

<i>Options</i>	<ul style="list-style-type: none"> <li>• <code>-cdi</code> The parsed CDI xml. Required and there is no default.</li> <li>• <code>-nid</code> The Node ID of the node to be configured. Required and there is no default.</li> <li>• <code>-transport</code> The transport object. Needs to implement <code>SendDatagram</code>, <code>DatagramReceivedOK</code>, and <code>DatagramRejected</code> methods and have an <code>-datagramhandler</code> option.</li> <li>• <code>-displayonly</code> A flag indicating that the CDI is just to be displayed. The default is false.</li> <li>• <code>-offlineedit</code> A flag indicating that the this is an offline editor. The default is false.</li> <li>• <code>-loadfile</code> The backup config filename to load from. Only checked if <code>-offlineedit</code> is true. Default is an empty string.</li> <li>• <code>-debugprint</code> A function to handle debug output.</li> <li>• <code>-class</code> Delegated to the toplevel.</li> <li>• <code>-menu</code> Delegated to the toplevel</li> <li>• <code>-height</code> Delegated to the ScrollableFrame</li> <li>• <code>-areaheight</code> Delegated to the ScrollableFrame</li> <li>• <code>-width</code> Delegated to the ScrollableFrame</li> <li>• <code>-areawidth</code> Delegated to the ScrollableFrame</li> </ul>
----------------	---

### 9.29.2 Constructor & Destructor Documentation

#### 9.29.2.1 ConfigurationEditor()

```
lcc::ConfigurationEditor::ConfigurationEditor (
    name ,
    ... )
```

Constructor: create the configuration editor.

Construct a memory configuration window to edit the configuration memory of an OpenLCB node. The window is created from the toplevel up.



## Parameters

<i>name</i>	Widget path.
...	<p>Options:</p> <ul style="list-style-type: none"> <li>• <code>-cdi</code> The parsed CDI xml. Required and there is no default.</li> <li>• <code>-nid</code> The Node ID of the node to be configured. Required and there is no default.</li> <li>• <code>-transport</code> The transport object. Needs to implement <code>SendDatagram</code>, <code>DatagramReceivedOK</code>, and <code>DatagramRejected</code> methods and have an <code>-datagramhandler</code> option.</li> <li>• <code>-displayonly</code> A flag indicating that the CDI is just to be displayed. The default is false.</li> <li>• <code>-offlineedit</code> A flag indicating that the this is an offline editor. The default is false.</li> <li>• <code>-loadfile</code> The backup config filename to load from. Only checked if <code>-offlineedit</code> is true. Default is an empty string.</li> <li>• <code>-debugprint</code> A function to handle debug output.</li> <li>• <code>-class</code> Delegated to the toplevel.</li> <li>• <code>-menu</code> Delegated to the toplevel</li> <li>• <code>-height</code> Delegated to the ScrollableFrame</li> <li>• <code>-areaheight</code> Delegated to the ScrollableFrame</li> <li>• <code>-width</code> Delegated to the ScrollableFrame</li> <li>• <code>-areawidth</code> Delegated to the ScrollableFrame</li> </ul>

## 9.29.3 Member Function Documentation

9.29.3.1 `_close()`

```
lcc::ConfigurationEditor::_close ( ) [private]
```

Close the window.

The window is withdrawn.

9.29.3.2 `_datagramhandler()`

```
lcc::ConfigurationEditor::_datagramhandler (
    command ,
    sourcenid ,
    ... ) [private]
```

Datagram handler.

## Parameters

<i>command</i>	Type of Datagram handling.
<i>sourcenid</i>	Source NID of the datagram.
...	The datagram data stream.

9.29.3.3 `_eventidComboRead()`

```
lcc::ConfigurationEditor::_eventidComboRead (
    widget ,
    space ,
    address ,
    size ) [private]
```

Read an event id value and map it to a ComboBox widget.

## Parameters

<i>widget</i>	A ttk::combobox widget to update. This is also used to map to the value map.
<i>space</i>	The space to read from.
<i>address</i>	The address of the event id.
<i>size</i>	The size of the event id (should always be 8).

9.29.3.4 `_eventidComboWrite()`

```
lcc::ConfigurationEditor::_eventidComboWrite (
    widget ,
    space ,
    address ,
    size ) [private]
```

Write an event id value mapped from a ComboBox widget.

## Parameters

<i>widget</i>	A ttk::combobox widget to get the value from. This is also used to map to the value map.
<i>space</i>	The space to read from.
<i>address</i>	The address of the event id.
<i>size</i>	The size of the event id (should always be 8).

## 9.29.3.5 \_eventidEntryRead()

```
lcc::ConfigurationEditor::_eventidEntryRead (
    widget ,
    space ,
    address ,
    size ) [private]
```

Read an event id value and stash it in an Entry widget as an event id string.

## Parameters

<i>widget</i>	A ttk::entry widget to update.
<i>space</i>	The space to read from.
<i>address</i>	The address of the event id.
<i>size</i>	The size of the event id (should always be 8).

## 9.29.3.6 \_eventidEntryWrite()

```
lcc::ConfigurationEditor::_eventidEntryWrite (
    widget ,
    space ,
    address ,
    size ) [private]
```

Write an event id value from an Entry widget.

## Parameters

<i>widget</i>	A ttk::entry widget to get the value from.
<i>space</i>	The space to read from.
<i>address</i>	The address of the event id.
<i>size</i>	The size of the event id (should always be 8).

## 9.29.3.7 \_intComboRead()

```
lcc::ConfigurationEditor::_intComboRead (
    widget ,
    space ,
    address ,
    size ) [private]
```

Read an integer value and map it to a ComboBox widget.

## Parameters

<i>widget</i>	A ttk::combobox widget to update. This is also used to map to the value map.
<i>space</i>	The space to read from.
<i>address</i>	The address of the integer.
<i>size</i>	The size of the integer.

## 9.29.3.8 \_intComboWrite()

```
lcc::ConfigurationEditor::_intComboWrite (
    widget ,
    space ,
    address ,
    size ,
    min ,
    max ) [private]
```

Write an integer value mapped from a ComboBox widget.

## Parameters

<i>widget</i>	A ttk::combobox widget to get the value from. This is also used to map to the value map.
<i>space</i>	The space to read from.
<i>address</i>	The address of the integer.
<i>size</i>	The size of the integer.
<i>min</i>	The minimum allowed value of the integer.
<i>max</i>	The maximum allowed value of the integer.

## 9.29.3.9 \_intSpinRead()

```
lcc::ConfigurationEditor::_intSpinRead (
    widget ,
    space ,
    address ,
    size ) [private]
```

Read an integer value and stash it in a SpinBox widget.

## Parameters

<i>widget</i>	A spinbox widget to update.
<i>space</i>	The space to read from.
<i>address</i>	The address of the integer.
<i>size</i>	The size of the integer.

## 9.29.3.10 \_intSpinWrite()

```
lcc::ConfigurationEditor::_intSpinWrite (
    widget ,
    space ,
    address ,
    size ,
    min ,
    max ) [private]
```

Write an integer value mapped from a SpinBox widget.

## Parameters

<i>widget</i>	A spinbox widget to get the value from.
<i>space</i>	The space to read from.
<i>address</i>	The address of the integer.
<i>size</i>	The size of the integer.
<i>min</i>	The minimum allowed value of the integer.
<i>max</i>	The maximum allowed value of the integer.

## 9.29.3.11 \_printexport()

```
lcc::ConfigurationEditor::_printexport (
    node ,
    frame ,
    name ) [private]
```

Print or export a segment or group.

The current contents of the specified segment or group GUI frame are exported to a data file for use in another program or printed.

## Parameters

<i>node</i>	The XML node in the CDI for the segment or group to export or print.
<i>frame</i>	The GUI frame containing the values to be exported or printed.
<i>name</i>	The name of the segment or group to be exported or printed.

### 9.29.3.12 \_printexport\_csv()

```
lcc::ConfigurationEditor::_printexport_csv (
    node ,
    frame ,
    name ,
    outfile ) [private]
```

Export a segment or group to a CSV file (can be imported into Excel).

#### Parameters

<i>node</i>	The XML node in the CDI for the segment or group to export or print.
<i>frame</i>	The GUI frame containing the values to be exported or printed.
<i>name</i>	The name of the segment or group to be exported or printed.
<i>outfile</i>	The file to export to.

### 9.29.3.13 \_printexport\_csv\_frame()

```
static lcc::ConfigurationEditor::_printexport_csv_frame (
    n ,
    matrix ,
    frame ) [static], [private]
```

Add a node's GUI frame values to a matrix (to be exported as a CSV file).

#### Parameters

<i>n</i>	The node in the CDI XML tree.
<i>matrix</i>	The matrix to populate.
<i>frame</i>	The GUI frame to extract values from.

### 9.29.3.14 \_printexport\_csv\_frameAcross()

```
static lcc::ConfigurationEditor::_printexport_csv_frameAcross (
    n ,
    rowVar ,
    frame ) [static], [private]
```

Add a group to a matrix as elements to a single row.

## Parameters

<i>n</i>	The node in the CDI XML tree.
<i>rowVar</i>	The name of the variable containing the row to add to.
<i>frame</i>	The GUI frame.

## 9.29.3.15 \_printexport\_csv\_framesAcross()

```
static lcc::ConfigurationEditor::_printexport_csv_framesAcross (
    n ,
    tabnb ,
    tabs ,
    matrix ) [static], [private]
```

Add a replicated group to a matrix as a single row.

## Parameters

<i>n</i>	The node in the CDI XML tree.
<i>tabnb</i>	Tabbed notebook containing the replicated group.
<i>tabs</i>	The tabs in the tabbed notebook (the replications).
<i>matrix</i>	The matrix to populate.

## 9.29.3.16 \_printexport\_csv\_vframe()

```
static lcc::ConfigurationEditor::_printexport_csv_vframe (
    n ,
    matrix ,
    frame ) [static], [private]
```

Add a scaler node's GUI value frame values to a matrix (to be exported as a CSV file).

## Parameters

<i>n</i>	The node in the CDI XML tree.
<i>matrix</i>	The matrix to populate
<i>frame</i>	The GUI frame to extract values from.

### 9.29.3.17 \_printexport\_csv\_vframeAcross()

```
static lcc::ConfigurationEditor::_printexport_csv_vframeAcross (
    n ,
    rowVar ,
    frame ) [static], [private]
```

Add a scaler node's value frame to a matrix as elements to a single row.

#### Parameters

<i>n</i>	The node in the CDI XML tree.
<i>rowVar</i>	The name of the variable containing the row to add to.
<i>frame</i>	The GUI frame.

### 9.29.3.18 \_printexport\_pdf()

```
lcc::ConfigurationEditor::_printexport_pdf (
    node ,
    frame ,
    name ,
    outfile ) [private]
```

Export a segment or group to a printable PDF file.

#### Parameters

<i>node</i>	The XML node in the CDI for the segment or group to export or print.
<i>frame</i>	The GUI frame containing the values to be exported or printed.
<i>name</i>	The name of the segment or group to be exported or printed.
<i>outfile</i>	The file to export to.

### 9.29.3.19 \_printexport\_pdf\_frame()

```
static lcc::ConfigurationEditor::_printexport_pdf_frame (
    n ,
    indent ,
    pdfobj ,
    frame ,
    curyVar ,
    curpageVar ,
    pageheader ) [static], [private]
```

Export a node frame to a PDF file.



## Parameters

<i>n</i>	The node.
<i>indent</i>	The indentation string.
<i>pdfobj</i>	The PDF file object.
<i>frame</i>	The GUI frame.
<i>curyVar</i>	The name of the variable containing the current y location.
<i>curpageVar</i>	The name of the variable containing the current page number.
<i>pageheader</i>	The running page header text.

## 9.29.3.20 \_printexport\_pdf\_newpage()

```
static lcc::ConfigurationEditor::_printexport_pdf_newpage (
    pdfobj ,
    pageheader ,
    pageno ) [static], [private]
```

Print a new PDF page.

## Parameters

<i>pdfobj</i>	The PDF file object
<i>pageheader</i>	The running page header text.
<i>pageno</i>	The new page's number.

## Returns

The fresh current y value.

## 9.29.3.21 \_printexport\_pdf\_vframe()

```
static lcc::ConfigurationEditor::_printexport_pdf_vframe (
    n ,
    indent ,
    pdfobj ,
    frame ,
    curyVar ,
    curpageVar ,
    pageheader ) [static], [private]
```

Export a node scaler value frame to a PDF file.

## Parameters

<i>n</i>	The node.
<i>indent</i>	The indentation string.
<i>pdfobj</i>	The PDF file object.
<i>frame</i>	The GUI frame.
<i>curyVar</i>	The name of the variable containing the current y location.
<i>curpageVar</i>	The name of the variable containing the current page number.
<i>pageheader</i>	The running page header text.

9.29.3.22 `_printexport_txt()`

```
lcc::ConfigurationEditor::_printexport_txt (
    node ,
    frame ,
    name ,
    outfile ) [private]
```

Export a segment or group to a text file.

## Parameters

<i>node</i>	The XML node in the CDI for the segment or group to export or print.
<i>frame</i>	The GUI frame containing the values to be exported or printed.
<i>name</i>	The name of the segment or group to be exported or printed.
<i>outfile</i>	The file to export to.

9.29.3.23 `_printexport_txt_frame()`

```
static lcc::ConfigurationEditor::_printexport_txt_frame (
    n ,
    indent ,
    outfp ,
    frame ) [static], [private]
```

Export a segment or group frame to a text file.

## Parameters

<i>n</i>	The node.
<i>indent</i>	The indentation string.
<i>outfp</i>	The output file channel.
<i>frame</i>	The GUI frame.

## 9.29.3.24 \_printexport\_txt\_vframe()

```
static lcc::ConfigurationEditor::_printexport_txt_vframe (
    n ,
    indent ,
    outfp ,
    frame ) [static], [private]
```

Export a node scaler value frame to a text file.

## Parameters

<i>n</i>	The node.
<i>indent</i>	The indentation string.
<i>outfp</i>	The output channel.
<i>frame</i>	The GUI frame.

## 9.29.3.25 \_printexport\_xml()

```
lcc::ConfigurationEditor::_printexport_xml (
    node ,
    frame ,
    name ,
    outfile ) [private]
```

Export a segment or group to an XML file.

## Parameters

<i>node</i>	The XML node in the CDI for the segment or group to export or print.
<i>frame</i>	The GUI frame containing the values to be exported or printed.
<i>name</i>	The name of the segment or group to be exported or printed.
<i>outfile</i>	The file to export to.

## 9.29.3.26 \_printexport\_xml\_frame()

```
static lcc::ConfigurationEditor::_printexport_xml_frame (
    n ,
    frame ) [static], [private]
```

Export a node frame as an XML tree.

## Parameters

<i>n</i>	The XML node in the CDI.
<i>frame</i>	The GUI frame for the node in the CDI.

## Returns

An XML tree of the contents of the GUI frame.

## 9.29.3.27 \_printexport\_xml\_vframe()

```
static lcc::ConfigurationEditor::_printexport_xml_vframe (
    n ,
    frame ) [static], [private]
```

Export a scaler node's value frame as an XML tree.

## Parameters

<i>n</i>	The XML node in the CDI.
<i>frame</i>	The GUI frame for the node in the CDI.

## Returns

An XML tree of the contents of the GUI frame.

## 9.29.3.28 \_processXMLnode()

```
lcc::ConfigurationEditor::_processXMLnode (
    n ,
    frame ,
    space ,
    address_var ,
    prefix = "" ) [private]
```

Process one node in the XML tree.

Process a single node in the XML tree. Will recurse to process Children nodes.

Ttk::labelframes are used for variables with names. Ttk::notebooks, except segments and groups. A ttk::labelframe is also used for the information block.

## Parameters

<i>n</i>	The node.
<i>frame</i>	The parent frame.
<i>space</i>	The current space.
<i>address_var</i>	The name of the address variable.

9.29.3.29 `_readall()`

```
lcc::ConfigurationEditor::_readall (
    space ) [private]
```

Read all parameters stored in a specified space.

Reads each parameter one at a time by invoking the parameter's `Read` button.

## Parameters

<i>space</i>	The parameter space to read from.
--------------	-----------------------------------

9.29.3.30 `_readmemory()`

```
lcc::ConfigurationEditor::_readmemory (
    space ,
    address ,
    length ,
    status_var ) [private]
```

Read memory from a space.

## Parameters

<i>space</i>	The space to read from.
<i>address</i>	The start address to read.
<i>length</i>	Number of bytes to read.
<i>status_var</i>	The name of a variable to receive the status code.

## Returns

The data read (if successful).

#### 9.29.3.31 \_stringComboRead()

```
lcc::ConfigurationEditor::_stringComboRead (
    widget ,
    space ,
    address ,
    size ) [private]
```

Read a string value and map it to a ComboBox widget.

##### Parameters

<i>widget</i>	A ttk::combobox widget to update. This is also used to map to the value map.
<i>space</i>	The space to read from.
<i>address</i>	The address of the string.
<i>size</i>	The size of the string.

#### 9.29.3.32 \_stringComboWrite()

```
lcc::ConfigurationEditor::_stringComboWrite (
    widget ,
    space ,
    address ,
    size ) [private]
```

Write a string value mapped from a ComboBox widget.

##### Parameters

<i>widget</i>	A ttk::combobox widget to get the value from. This is also used to map to the value map.
<i>space</i>	The space to read from.
<i>address</i>	The address of the string.
<i>size</i>	The size of the string.

#### 9.29.3.33 \_stringEntryRead()

```
lcc::ConfigurationEditor::_stringEntryRead (
    widget ,
    space ,
    address ,
    size ) [private]
```

Read a string value and stash it in an Entry widget.

## Parameters

<i>widget</i>	A ttk::entry widget to update.
<i>space</i>	The space to read from.
<i>address</i>	The address of the string.
<i>size</i>	The size of the string.

## 9.29.3.34 \_stringEntryWrite()

```
lcc::ConfigurationEditor::_stringEntryWrite (
    widget ,
    space ,
    address ,
    size ) [private]
```

Write a string value from an Entry widget.

## Parameters

<i>widget</i>	A ttk::entry widget to get the value from.
<i>space</i>	The space to read from.
<i>address</i>	The address of the string.
<i>size</i>	The size of the string.

## 9.29.3.35 \_writememory()

```
lcc::ConfigurationEditor::_writememory (
    space ,
    address ,
    databuffer ) [private]
```

Write to configuration memory.

## Parameters

<i>space</i>	The space to write to.
<i>address</i>	The address to write to.
<i>databuffer</i>	The data to write.



**Returns**

The write status.

**9.29.3.36 putdebug()**

```
lcc::ConfigurationEditor::putdebug (
    message ) [private]
```

Print message using debug output, if any.

**Parameters**

<i>message</i>	The message to print.
----------------	-----------------------

**9.29.4 Member Data Documentation****9.29.4.1 \_datagramrejecterror**

```
lcc::ConfigurationEditor::_datagramrejecterror [private]
```

Datagram reject error flag.

**9.29.4.2 \_eventidnumber**

```
lcc::ConfigurationEditor::_eventidnumber [private]
```

Eventid number, used to insure unique widget names.

**9.29.4.3 \_groupnumber**

```
lcc::ConfigurationEditor::_groupnumber [private]
```

Group number, used to insure unique widget names.

#### 9.29.4.4 `_intnumber`

```
lcc::ConfigurationEditor::_intnumber [private]
```

Integer number, used to insure unique widget names.

#### 9.29.4.5 `_ioComplete`

```
lcc::ConfigurationEditor::_ioComplete [private]
```

I/O Completion Flag.

#### 9.29.4.6 `_menu`

```
lcc::ConfigurationEditor::_menu [static], [private]
```

Generic menu.

#### 9.29.4.7 `_mkbuttons`

```
lcc::ConfigurationEditor::_mkbuttons [private]
```

Flag for Make Sensor / Make Turnout etc.

buttons

#### 9.29.4.8 `_readall`

```
lcc::ConfigurationEditor::_readall [private]
```

Holds all of the Read buttons for each segment.

This allows for Reading all of the variables in a segment.

#### 9.29.4.9 `_segmentnumber`

```
lcc::ConfigurationEditor::_segmentnumber [private]
```

Segment number, used to insure unique widget names.

#### 9.29.4.10 \_stringnumber

```
lcc::ConfigurationEditor::_stringnumber [private]
```

String number, used to insure unique widget names.

#### 9.29.4.11 buttons

```
lcc::ConfigurationEditor::buttons [private]
```

Button box.

#### 9.29.4.12 cdi

```
lcc::ConfigurationEditor::cdi [private]
```

CDI XML Object.

#### 9.29.4.13 datagrambuffer

```
lcc::ConfigurationEditor::datagrambuffer [private]
```

Datagram buffer.

#### 9.29.4.14 editframe

```
lcc::ConfigurationEditor::editframe [private]
```

Scrollable Frame.

#### 9.29.4.15 idheaders

```
lcc::ConfigurationEditor::idheaders [static], [private]
```

Locale versions of the identification headers.

**9.29.4.16 main**

```
lcc::ConfigurationEditor::main [private]
```

Main Frame.

**9.29.4.17 olddatagramhandler**

```
lcc::ConfigurationEditor::olddatagramhandler [private]
```

Variable holding the old Datagram handler.

**9.29.4.18 printexportfiletypes**

```
lcc::ConfigurationEditor::printexportfiletypes [static], [private]
```

Print and Export file types.

**9.29.4.19 scroll**

```
lcc::ConfigurationEditor::scroll [private]
```

Scrolled Window.

**9.29.4.20 statusline**

```
lcc::ConfigurationEditor::statusline [private]
```

Status variable.

**9.29.4.21 writeReplyCheck**

```
lcc::ConfigurationEditor::writeReplyCheck [private]
```

Datagram write trply check flag.

## 9.30 Parsers::CornuBody Class Reference

List of Cornu body lines (T, E, S, and C lines).

```
#include <CornuBody.h>
```

### Public Member Functions

- [CornuBody](#) ([CornuBodyElt](#) \*e, [CornuBody](#) \*n)  
*Basic constructor.*
- [TrackBody](#) \* [CornuEnds](#) ()  
*Create a track endpoint list.*
- int [CornuSegmentCount](#) ()  
*Count segments (S, C, and J lines).*
- const [CornuBodyElt](#) \* [Element](#) () const  
*Return current element.*

### Static Public Member Functions

- static [CornuBody](#) \* [ConsCornuBody](#) ([CornuBodyElt](#) \*trbe, [CornuBody](#) \*trb)  
*Alternative constructor function.*
- static [CornuBody](#) \* [ConcatCornuBody](#) ([CornuBody](#) \*trba, [CornuBody](#) \*trb)
- static void [CleanUpCornuBody](#) ([CornuBody](#) \*trb)  
*Free up memory.*

### Private Member Functions

- void [CleanUpElement](#) ()  
*Free up memory.*

### Private Attributes

- [CornuBodyElt](#) \* [element](#)  
*Current element.*
- [CornuBody](#) \* [next](#)  
*Next element.*

### Friends

- class [CornuBodyElt](#)
- class [TrackGraph](#)

### 9.30.1 Detailed Description

List of Cornu body lines (T, E, S, and C lines).

#### Author

Robert Heller <heller@deepsoft.com>

### 9.30.2 Constructor & Destructor Documentation

#### 9.30.2.1 CornuBody()

```
Parsers::CornuBody::CornuBody (  
    CornuBodyElt * e,  
    CornuBody * n ) [inline]
```

Basic constructor.

### 9.30.3 Member Function Documentation

#### 9.30.3.1 CleanUpCornuBody()

```
static void Parsers::CornuBody::CleanUpCornuBody (  
    CornuBody * trb ) [inline], [static]
```

Free up memory.

References CleanUpElement(), element, and next.

#### 9.30.3.2 CleanUpElement()

```
void Parsers::CornuBody::CleanUpElement ( ) [inline], [private]
```

Free up memory.

References Parsers::CornuBodyElt::theEnd, and Parsers::CornuBodyElt::theType.

Referenced by CleanUpCornuBody().

### 9.30.3.3 ConcatCornuBody()

```
static CornuBody* Parsers::CornuBody::ConcatCornuBody (
    CornuBody * trba,
    CornuBody * trb ) [inline], [static]
```

References next.

### 9.30.3.4 ConsCornuBody()

```
static CornuBody* Parsers::CornuBody::ConsCornuBody (
    CornuBodyElt * trbe,
    CornuBody * trb ) [inline], [static]
```

Alternative constructor function.

### 9.30.3.5 CornuEnds()

```
TrackBody* Parsers::CornuBody::CornuEnds ( ) [inline]
```

Create a track endpoint list.

References element, next, Parsers::CornuBodyElt::theEnd, and Parsers::CornuBodyElt::theType.

### 9.30.3.6 CornuSegmentCount()

```
int Parsers::CornuBody::CornuSegmentCount ( ) [inline]
```

Count segments (S, C, and J lines).

References element, next, and Parsers::CornuBodyElt::theType.

### 9.30.3.7 Element()

```
const CornuBodyElt* Parsers::CornuBody::Element ( ) const [inline]
```

Return current element.

### 9.30.4 Friends And Related Function Documentation

#### 9.30.4.1 CornuBodyElt

```
friend class CornuBodyElt [friend]
```

#### 9.30.4.2 TrackGraph

```
friend class TrackGraph [friend]
```

### 9.30.5 Member Data Documentation

#### 9.30.5.1 element

```
CornuBodyElt* Parsers::CornuBody::element [private]
```

Current element.

Referenced by `CleanUpCornuBody()`, `CornuEnds()`, and `CornuSegmentCount()`.

#### 9.30.5.2 next

```
CornuBody* Parsers::CornuBody::next [private]
```

Next element.

Referenced by `CleanUpCornuBody()`, `ConcatCornuBody()`, `CornuEnds()`, and `CornuSegmentCount()`.

## 9.31 Parsers::CornuBodyElt Class Reference

Cornu Body elements: T, E, S, and C lines are collected.

```
#include <CornuBody.h>
```



## Classes

- struct [Pos](#)  
*Position structure.*

## Public Types

- enum [CornuBodyEltType](#) { [None](#), [CornuEnd](#), [CornuStraightSegment](#), [CornuCurvedSegment](#) }  
*Element types.*

## Public Member Functions

- [CornuBodyElt](#) ()  
*Constructor.*
- [~CornuBodyElt](#) ()  
*Destructor.*
- [CornuBodyEltType TheType](#) () const  
*Type accessor.*
- int [GetStraightSegment](#) (float &x1, float &y1, float &x2, float &y2) const  
*Fetch straight segment data.*
- int [GetCurveSegment](#) (float &r, float &x, float &y, float &a0, float &a1) const  
*Fetch curve segment data.*

## Static Public Member Functions

- static void [InitTSegId](#) ()  
*Segment count initializer.*
- static [CornuBodyElt](#) \* [MakeTrackEnd](#) ([TrackBodyElt](#) \*tbe)  
*Create an endpoint (T or E lines).*
- static [CornuBodyElt](#) \* [MakeStraightSegment](#) (float x1, float y1, float x2, float y2)  
*Create a straight segment (S lines).*
- static [CornuBodyElt](#) \* [MakeCurveSegment](#) (float r, float x, float y, float a0, float a1)  
*Create a curve segment (C lines).*

## Private Attributes

- [CornuBodyEltType theType](#)  
*Element type.*
- [TrackBodyElt](#) \* [theEnd](#)  
*Pointer to T or E line data.*
- int [segmentId](#)  
*Segment index (S or C lines).*
- [Pos pos1](#)  
*First position.*

- [Pos pos2](#)  
*Second position.*
- float [radius](#)  
*A radius value.*
- float [ang0](#)  
*An angle value.*
- float [ang1](#)  
*Another angle value.*

### Static Private Attributes

- static int [segCount](#)  
*Counter for S and C segments.*

### Friends

- class [TrackGraph](#)
- class [CornuBody](#)

## 9.31.1 Detailed Description

Cornu Body elements: T, E, S, and C lines are collected.

others are discarded.

#### Author

Robert Heller <heller@deepsoft.com>

## 9.31.2 Member Enumeration Documentation

### 9.31.2.1 CornuBodyEltType

enum [Parsers::CornuBodyElt::CornuBodyEltType](#)

Element types.

#### Enumerator

None	Placeholder.
CornuEnd	T or E line.
CornuStraightSegment	S line.
CornuCurvedSegment	C line.

### 9.31.3 Constructor & Destructor Documentation

#### 9.31.3.1 CornuBodyElt()

```
Parsers::CornuBodyElt::CornuBodyElt ( ) [inline]
```

Constructor.

#### 9.31.3.2 ~CornuBodyElt()

```
Parsers::CornuBodyElt::~~CornuBodyElt ( ) [inline]
```

Destructor.

### 9.31.4 Member Function Documentation

#### 9.31.4.1 GetCurveSegment()

```
int Parsers::CornuBodyElt::GetCurveSegment (
    float & r,
    float & x,
    float & y,
    float & a0,
    float & a1 ) const [inline]
```

Fetch curve segment data.

References Parsers::CornuBodyElt::Pos::x, and Parsers::CornuBodyElt::Pos::y.

#### 9.31.4.2 GetStraightSegment()

```
int Parsers::CornuBodyElt::GetStraightSegment (
    float & x1,
    float & y1,
    float & x2,
    float & y2 ) const [inline]
```

Fetch straight segment data.

References Parsers::CornuBodyElt::Pos::x, and Parsers::CornuBodyElt::Pos::y.

#### 9.31.4.3 InitTSegId()

```
static void Parsers::CornuBodyElt::InitTSegId ( ) [inline], [static]
```

Segment count initializer.

#### 9.31.4.4 MakeCurveSegment()

```
static CornuBodyElt* Parsers::CornuBodyElt::MakeCurveSegment (
    float r,
    float x,
    float y,
    float a0,
    float a1 ) [inline], [static]
```

Create a curve segment (C lines).

References ang0, ang1, pos1, radius, segmentId, theType, Parsers::CornuBodyElt::Pos::x, and Parsers::CornuBodyElt::Pos::y.

#### 9.31.4.5 MakeStraightSegment()

```
static CornuBodyElt* Parsers::CornuBodyElt::MakeStraightSegment (
    float x1,
    float y1,
    float x2,
    float y2 ) [inline], [static]
```

Create a straight segment (S lines).

References pos1, pos2, segmentId, theType, Parsers::CornuBodyElt::Pos::x, and Parsers::CornuBodyElt::Pos::y.

#### 9.31.4.6 MakeTrackEnd()

```
static CornuBodyElt* Parsers::CornuBodyElt::MakeTrackEnd (
    TrackBodyElt * tbe ) [inline], [static]
```

Create an endpoint (T or E lines).

References theEnd, and theType.

#### 9.31.4.7 TheType()

```
CornuBodyEltType Parsers::CornuBodyElt::TheType ( ) const [inline]
```

Type accessor.

### 9.31.5 Friends And Related Function Documentation

#### 9.31.5.1 CornuBody

```
friend class CornuBody [friend]
```

#### 9.31.5.2 TrackGraph

```
friend class TrackGraph [friend]
```

### 9.31.6 Member Data Documentation

#### 9.31.6.1 ang0

```
float Parsers::CornuBodyElt::ang0 [private]
```

An angle value.

Referenced by MakeCurveSegment().

#### 9.31.6.2 ang1

```
float Parsers::CornuBodyElt::ang1 [private]
```

Another angle value.

Referenced by MakeCurveSegment().

#### 9.31.6.3 pos1

```
Pos Parsers::CornuBodyElt::pos1 [private]
```

First position.

Referenced by MakeCurveSegment(), and MakeStraightSegment().

#### 9.31.6.4 pos2

```
Pos Parsers::CornuBodyElt::pos2 [private]
```

Second position.

Referenced by MakeStraightSegment().

#### 9.31.6.5 radius

```
float Parsers::CornuBodyElt::radius [private]
```

A radius value.

Referenced by MakeCurveSegment().

#### 9.31.6.6 segCount

```
int Parsers::CornuBodyElt::segCount [static], [private]
```

Counter for S and C segments.

#### 9.31.6.7 segmentId

```
int Parsers::CornuBodyElt::segmentId [private]
```

Segment index (S or C lines).

Referenced by MakeCurveSegment(), and MakeStraightSegment().

## 9.31.6.8 theEnd

`TrackBodyElt* Parsers::CornuBodyElt::theEnd [private]`

Pointer to T or E line data.

Referenced by `Parsers::CornuBody::CleanUpElement()`, `Parsers::CornuBody::CornuEnds()`, and `MakeTrackEnd()`.

## 9.31.6.9 theType

`CornuBodyEltType Parsers::CornuBodyElt::theType [private]`

Element type.

Referenced by `Parsers::CornuBody::CleanUpElement()`, `Parsers::CornuBody::CornuEnds()`, `Parsers::CornuBody::↔CornuSegmentCount()`, `MakeCurveSegment()`, `MakeStraightSegment()`, and `MakeTrackEnd()`.

## 9.32 CTCPanel::Crossing Class Reference

Crossing object type.

## Public Member Functions

- `Crossing` (name, \_ctcpanel, \_canvas,...)  
*Construct a `Crossing` object.*
- `~Crossing` ()  
*Clean up all data objects and free up all resources.*
- `getv` ()  
*Method to get our value (state).*
- `setv` (value)  
*Method to set out value (state).*
- `geti` (ind)  
*Method to get the state of one of our indicators (none).*
- `seti` (ind, value)  
*Method to set an indicator's state (none).*
- `invoke` ()  
*Method to invoke the switch.*

## Private Member Functions

- `_configureLabel` (option, value)  
*Method to update the label option.*
- `_VerifyCrossingType` (option, value)

## Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*

### 9.32.1 Detailed Description

Crossing object type.

These are on the schematic and represent a piece of track on the Schematic.

#### Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the switch on.
...	Options: <ul style="list-style-type: none"> <li>• <code>-x</code> The x coordinate of the object (readonly, default 0).</li> <li>• <code>-y</code> The y coordinate of the object (readonly, default 0).</li> <li>• <code>-controlpoint</code> The name of the control point this label is part of (readonly, default CP1).</li> <li>• <code>-label</code> The label of the switch (default "1").</li> <li>• <code>-orientation</code> The orientation (8-way) of the switch (readonly, default 0).</li> <li>• <code>-flipped</code> Whether or not the switch is flipped (readonly, default no).</li> <li>• <code>-type</code> The type of crossing (x90 or x45) (readonly, default x90).</li> <li>• <code>-occupiedcommand</code> A command to run to find out if the switch is occupied (default {}).</li> </ul>

Defined coords terminals:

- MainL Mainline left.
- MainR Mainline right.
- AltL Alternative line left.
- AltR Alternative line right.

Defined values (states): none. Defined indicators: none.

#### Author

Robert Heller <heller@deepsoft.com>



## 9.32.2 Constructor & Destructor Documentation

### 9.32.2.1 Crossing()

```
CTCPanel::Crossing::Crossing (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a [Crossing](#) object.

#### Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the <a href="#">Crossing</a> on.
<code>...</code>	Option list.

### 9.32.2.2 ~Crossing()

```
CTCPanel::Crossing::~~Crossing ( )
```

Clean up all data objects and free up all resources.

## 9.32.3 Member Function Documentation

### 9.32.3.1 \_configureLabel()

```
CTCPanel::Crossing::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

#### 9.32.3.2 `_VerifyCrossingType()`

```
CTCPanel::Crossing::_VerifyCrossingType (
    option ,
    value ) [private]
```

#### 9.32.3.3 `geti()`

```
CTCPanel::Crossing::geti (
    ind )
```

Method to get the state of one of our indicators (none).

#### 9.32.3.4 `getv()`

```
CTCPanel::Crossing::getv ( )
```

Method to get our value (state).

#### 9.32.3.5 `invoke()`

```
CTCPanel::Crossing::invoke ( )
```

Method to invoke the switch.

#### 9.32.3.6 `seti()`

```
CTCPanel::Crossing::seti (
    ind ,
    value )
```

Method to set an indicator's state (none).

#### 9.32.3.7 `setv()`

```
CTCPanel::Crossing::setv (
    value )
```

Method to set out value (state).

## Parameters

<i>value</i>	The new state to set.
--------------	-----------------------

## 9.32.4 Member Data Documentation

## 9.32.4.1 canvas

CTCPanel::Crossing::canvas [private]

The canvas component (parent widget component).

## 9.32.4.2 ctcpnl

CTCPanel::Crossing::ctcpnl [private]

The CTC Panel component (parent widget).

## 9.33 CTCPanel::Crossover Class Reference

[Crossover](#) (turnout) object type.

## Public Member Functions

- [Crossover](#) (name, \_ctcpnl, \_canvas,...)  
*Construct a [Crossover](#) object.*
- [~Crossover](#) ()  
*Clean up all data objects and free up all resources.*
- [getv](#) ()  
*Method to get our value (state).*
- [setv](#) (value)  
*Method to set out value (state).*
- [geti](#) (ind)  
*Method to get the state of one of our indicators (none).*
- [seti](#) (ind, value)  
*Method to set an indicator's state (none).*
- [invoke](#) ()  
*Method to invoke the switch.*

## Private Member Functions

- [\\_configureLabel](#) (option, value)  
*Method to update the label option.*

## Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*
- [state](#)  
*The state of the points.*

### 9.33.1 Detailed Description

[Crossover](#) (turnout) object type.

These are on the schematic and represent a [Crossover](#) on the Schematic.

#### Parameters

<a href="#">_ctcpanel</a>	The <a href="#">CTCPanel</a> megawidget.
<a href="#">_canvas</a>	The schematic canvas to draw the switch on.
...	Options: <ul style="list-style-type: none"> <li>• -x The x coordinate of the object (readonly, default 0).</li> <li>• -y The y coordinate of the object (readonly, default 0).</li> <li>• -controlpoint The name of the control point this label is part of (readonly, default CP1).</li> <li>• -label The label of the switch (default "1").</li> <li>• -orientation The orientation (8-way) of the switch (readonly, default 0).</li> <li>• -flipped Whether or not the switch is flipped (readonly, default no).</li> <li>• -statecommand A command to run to get the switch's state (default {}).</li> <li>• -occupiedcommand A command to run to find out if the switch is occupied (default {}).</li> </ul>

Defined coords terminals:

- Main1L Upper left mainline.
- Main2L Lower left mainline.
- Main1R Upper right mainline.
- Main2R Lower right mainline.

Defined values (states):

- Normal Points are aligned for the mainline.
- Reverse Points are aligned for the branchline.
- Unknown Point are not aligned for any route (eg points are in motion).

Defined indicators: none.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.33.2 Constructor & Destructor Documentation

#### 9.33.2.1 Crossover()

```
CTCPanel::Crossover::Crossover (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a [Crossover](#) object.

##### Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the <a href="#">Crossover</a> on.
<code>...</code>	Option list.

#### 9.33.2.2 ~Crossover()

```
CTCPanel::Crossover::~~Crossover ( )
```

Clean up all data objects and free up all resources.

### 9.33.3 Member Function Documentation

#### 9.33.3.1 `_configureLabel()`

```
CTCPanel::Crossover::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

#### 9.33.3.2 `geti()`

```
CTCPanel::Crossover::geti (
    ind )
```

Method to get the state of one of our indicators (none).

#### 9.33.3.3 `getv()`

```
CTCPanel::Crossover::getv ( )
```

Method to get our value (state).

#### 9.33.3.4 `invoke()`

```
CTCPanel::Crossover::invoke ( )
```

Method to invoke the switch.

#### 9.33.3.5 `seti()`

```
CTCPanel::Crossover::seti (
    ind ,
    value )
```

Method to set an indicator's state (none).

#### 9.33.3.6 `setv()`

```
CTCPanel::Crossover::setv (
    value )
```

Method to set out value (state).

## Parameters

<i>value</i>	The new state to set.
--------------	-----------------------

### 9.33.4 Member Data Documentation

#### 9.33.4.1 canvas

```
CTCPanel::Crossover::canvas [private]
```

The canvas component (parent widget component).

#### 9.33.4.2 ctcpnl

```
CTCPanel::Crossover::ctcpnl [private]
```

The CTC Panel component (parent widget).

#### 9.33.4.3 state

```
CTCPanel::Crossover::state [private]
```

The state of the points.

## 9.34 FCFSupport::PDFFileStructures::CrossReferenceTable Class Reference

The cross reference table object.

```
#include <PDFPrinterSupport.h>
```

## Public Member Functions

- [CrossReferenceTable](#) ()  
*Constructor.*
- [~CrossReferenceTable](#) ()  
*Destructor.*
- void [AddIndirectObjectToTable](#) ([IndirectObject](#) \*obj)  
*Add an indirect object to the cross reference table.*
- streampos [WriteTable](#) (ostream &stream) const  
*Write this cross reference table out.*
- unsigned long int [HighestObjectNumber](#) () const  
*Return the highest object number.*

## Private Types

- typedef map< unsigned long int, [IndirectObject](#) \*, less< unsigned long int > > [objectMap](#)  
*The object table type.*

## Private Member Functions

- void [FreeObject](#) (unsigned long int objNum)  
*Free up a object slot in the cross reference table.*

## Private Attributes

- [objectMap](#) [objectTable](#)  
*The table of objects.*

## Static Private Attributes

- static unsigned long int [lastObjectNumber](#)  
*The last used object number.*

## Friends

- class [IndirectObject](#)

### 9.34.1 Detailed Description

The cross reference table object.

The stricture holds the cross-reference table, which is used to index and access indirect objects of various sorts.

#### Author

Robert Heller <heller@deepsoft.com>



## 9.34.2 Member Typedef Documentation

### 9.34.2.1 objectMap

```
typedef map<unsigned long int, IndirectObject*, less<unsigned long int> > FCFSupport::PDFFileStructures::CrossReferenceTable::objectMap [private]
```

The object table type.

## 9.34.3 Constructor & Destructor Documentation

### 9.34.3.1 CrossReferenceTable()

```
FCFSupport::PDFFileStructures::CrossReferenceTable::CrossReferenceTable ( )
```

Constructor.

Initialize a cross reference table object.

### 9.34.3.2 ~CrossReferenceTable()

```
FCFSupport::PDFFileStructures::CrossReferenceTable::~~CrossReferenceTable ( ) [inline]
```

Destructor.

Cleans things up.

References AddIndirectObjectToTable(), lcc::stream, and WriteTable().

## 9.34.4 Member Function Documentation

### 9.34.4.1 AddIndirectObjectToTable()

```
void FCFSupport::PDFFileStructures::CrossReferenceTable::AddIndirectObjectToTable (
    IndirectObject * obj )
```

Add an indirect object to the cross reference table.

**Parameters**

<i>obj</i>	The object to add.
------------	--------------------

Referenced by `~CrossReferenceTable()`.

**9.34.4.2 FreeObject()**

```
void FCFSupport::PDFFileStructures::CrossReferenceTable::FreeObject (
    unsigned long int objNum ) [private]
```

Free up a object slot in the cross reference table.

**Parameters**

<i>objNum</i>	The object number to free up.
---------------	-------------------------------

Referenced by `HighestObjectNumber()`.

**9.34.4.3 HighestObjectNumber()**

```
unsigned long int FCFSupport::PDFFileStructures::CrossReferenceTable::HighestObjectNumber ( )
const [inline]
```

Return the highest object number.

References `FreeObject()`, and `lastObjectNumber`.

**9.34.4.4 WriteTable()**

```
streampos FCFSupport::PDFFileStructures::CrossReferenceTable::WriteTable (
    ostream & stream ) const
```

Write this cross reference table out.

Returns the file position.

**Parameters**

<i>stream</i>	The stream to write to.
---------------	-------------------------

Referenced by `~CrossReferenceTable()`.

### 9.34.5 Friends And Related Function Documentation

#### 9.34.5.1 IndirectObject

```
friend class IndirectObject [friend]
```

### 9.34.6 Member Data Documentation

#### 9.34.6.1 lastObjectNumber

```
unsigned long int FCFSupport::PDFFileStructures::CrossReferenceTable::lastObjectNumber [static],  
[private]
```

The last used object number.

Referenced by `HighestObjectNumber()`.

#### 9.34.6.2 objectTable

```
objectMap FCFSupport::PDFFileStructures::CrossReferenceTable::objectTable [private]
```

The table of objects.

## 9.35 CTCPanel::CTCLabel Class Reference

CTC Label object type.

## Public Member Functions

- [CTCLabel](#) (name, \_ctcpanel, \_canvas,...)  
*Construct a Label object.*
- [~CTCLabel](#) ()  
*Clean up all data objects and free up all resources.*
- [getv](#) ()  
*Method to get our value (none).*
- [setv](#) (state)  
*Method to set out value (level position).*
- [geti](#) (ind)  
*Method to get the state of one of our indicators (none).*
- [seti](#) (ind, value)  
*Method to set an indicator's state (none).*
- [invoke](#) ()  
*Method to invoke the label.*

## Private Member Functions

- [\\_configureColor](#) (option, value)  
*Method to update the color of the label.*
- [\\_configureLabel](#) (option, value)  
*Method to update the label option.*

## Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*

## 9.35.1 Detailed Description

CTC Label object type.

These are on the control panel and represent a label on the CTC Panel

### Parameters

<a href="#">_ctcpanel</a>	The <a href="#">CTCPanel</a> megawidget.
<a href="#">_canvas</a>	The control panel canvas to draw the label on.
...	Options: <ul style="list-style-type: none"> <li>• -x The x coordinate of the object (readonly, default 0).</li> <li>• -y The y coordinate of the object (readonly, default 0).</li> <li>• <del>-controlpoint</del> The name of the control point this label is part of (readonly, default CP1).</li> <li>• -color The color of the label (default white).</li> <li>• -label The label of the label (default "").</li> </ul>

Defined coords terminals: none. Defined values (states): none. Defined indicators: none.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.35.2 Constructor & Destructor Documentation

#### 9.35.2.1 CTCLabel()

```
CTCPanel::CTCLabel::CTCLabel (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a Label object.

##### Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The control panel canvas to draw the <a href="#">CTCLabel</a> on.
<code>...</code>	Option list.

#### 9.35.2.2 ~CTCLabel()

```
CTCPanel::CTCLabel::~~CTCLabel ( )
```

Clean up all data objects and free up all resources.

### 9.35.3 Member Function Documentation

#### 9.35.3.1 `_configureColor()`

```
CTCPanel::CTCLabel::_configureColor (
    option ,
    value ) [private]
```

Method to update the color of the label.

#### 9.35.3.2 `_configureLabel()`

```
CTCPanel::CTCLabel::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

#### 9.35.3.3 `geti()`

```
CTCPanel::CTCLabel::geti (
    ind )
```

Method to get the state of one of our indicators (none).

#### 9.35.3.4 `getv()`

```
CTCPanel::CTCLabel::getv ( )
```

Method to get our value (none).

#### 9.35.3.5 `invoke()`

```
CTCPanel::CTCLabel::invoke ( )
```

Method to invoke the label.

#### 9.35.3.6 seti()

```
CTCPanel::CTCLabel::seti (
    ind ,
    value )
```

Method to set an indicator's state (none).

#### 9.35.3.7 setv()

```
CTCPanel::CTCLabel::setv (
    state )
```

Method to set out value (level position).

### 9.35.4 Member Data Documentation

#### 9.35.4.1 canvas

```
CTCPanel::CTCLabel::canvas [private]
```

The canvas component (parent widget component).

#### 9.35.4.2 ctcpnl

```
CTCPanel::CTCLabel::ctcpnl [private]
```

The CTC Panel component (parent widget).

## 9.36 CTCPanel::CTCPanel Class Reference

Main CTC Panel megawidget.

## Public Member Functions

- [CTCPanel](#) (name,...)  
*Build and install all component widgets and process configuration.*
- [updateSR](#) (canvas, newheight, newwidth)  
*Method to update one of the canvases scroll region.*
- [zoomBy](#) (zoomFactor)  
*Method to zoom the display by a zoom factor.*
- [setZoom](#) (zoomFactor)  
*Method to set the zoom factor to a specific factor.*
- [getZoom](#) ()  
*Return the zoom (scaling) factor.*
- [getv](#) (name)  
*Method to get the value (or state) of an object.*
- [setv](#) (name, value)  
*Method to set the value (or state) of an object.*
- [geti](#) (name, ind)  
*Method to get the indicator state of an object.*
- [seti](#) (name, ind, value)  
*Method to set the indicator state of an object.*
- [itemcget](#) (name, option)  
*Method to get a configuration option from an object.*
- [itemconfigure](#) (name,...)  
*Method to set a configuration option from an object.*
- [exists](#) (name)  
*Test if the named object exists.*
- [delete](#) (name)  
*Delete a named object.*
- [move](#) (name, x, y)  
*Move a named object.*
- [invoke](#) (name)  
*Method to invoke an object.*
- [coords](#) (name, tname)  
*Method to fetch the coordinates of some part of an object.*
- [bind](#) (name,...)  
*Method to set a binding on an Object.*
- [print](#) (name, fp)  
*Method to print the named object to the specified file channel.*
- [objectlist](#) (cp="", class="")  
*Method to return the list of objects.*
- [cplist](#) ()  
*Method to return the list of controlpoints.*
- [create\\_SWPlate](#) (name,...)  
*Method to create a switch plate object.*
- [create\\_SIGPlate](#) (name,...)  
*Method to create a signal plate object.*
- [create\\_CodeButton](#) (name,...)



*Method to create a code button object.*

- [create\\_Toggle](#) (name,...)

*Method to create a toggle switch object.*

- [create\\_Lamp](#) (name,...)

*Method to create a lamp object.*

- [create\\_CTCLabel](#) (name,...)

*Method to create a CTC Label label object.*

- [create\\_Switch](#) (name,...)

*Method to create a switch (turnout) object.*

- [create\\_SchLabel](#) (name,...)

*Method to create a schematic label object.*

- [create\\_StraightBlock](#) (name,...)

*Method to create a straight block of track object.*

- [create\\_EndBumper](#) (name,...)

*Method to create a end bumper of track object.*

- [create\\_CurvedBlock](#) (name,...)

*Method to create a curved block of track object.*

- [create\\_ScissorCrossover](#) (name,...)

*Method to create a scissor crossover object.*

- [create\\_Crossover](#) (name,...)

*Method to create a crossover object.*

- [create\\_Crossing](#) (name,...)

*Method to create a track crossing object.*

- [create\\_SingleSlip](#) (name,...)

*Method to create a single slip object.*

- [create\\_DoubleSlip](#) (name,...)

*Method to create a double slip object.*

- [create\\_ThreeWaySW](#) (name,...)

*Method to create a three way switch object.*

- [create\\_HiddenBlock](#) (name,...)

*Method to create a hidden block of track object.*

- [create\\_StubYard](#) (name,...)

*Method to create a stub (deadend) yard object.*

- [create\\_ThroughYard](#) (name,...)

*Method to create a through yard object.*

- [create\\_PushButton](#) (name,...)

*Method to create a push button object.*

- [create\\_Signal](#) (name,...)

*Method to create a signal object.*

- [schematic\\_crosshair](#) (...)

*Display crosshairs on the schematic canvas.*

- [controls\\_crosshair](#) (...)

*Display crosshairs on the controls canvas.*

## Protected Member Functions

- [checkInitCP](#) (cp)  
*Method to check that a control point has been initialized.*
- [updateAndSyncCP](#) (cp)  
*Method to update and synchronize a control point.*
- [lappendCP](#) (cp, slot, what)  
*Method to lappend something to a slot in a control point's data.*
- [lremoveCP](#) (cp, slot, what)  
*Method to remove something from a slot in a control point's data.*

## Private Member Functions

- [\\_PosInteger](#) (option, value)  
*Method to validate a positive non zero integer option.*
- [\\_CtcMainSyncX](#) (this, other, first, last)  
*Internal method to x scroll updates.*
- [\\_CtcMainHScroll2](#) (...)  
*Internal method to scroll two canvases at the same time.*
- [\\_crosshairStart](#) (canvas, xvar, yvar)  
*Start using the crosshairs.*
- [\\_crosshairMove](#) (canvas, xvar, yvar, mx, my)  
*Bound to mouse movement events.*
- [\\_crosshairEnd](#) (canvas, xvar, yvar, mx, my)  
*Bound to left button down.*

## Private Attributes

- [schematic](#)  
*The schematic component.*
- [schematicYscroll](#)  
*The schematic y scroll bar component.*
- [xscroll](#)  
*The shared x scroll bar component.*
- [controls](#)  
*The controls component.*
- [controlsYscroll](#)  
*The controls y scroll bar component.*
- [scale](#)  
*The current scale value.*
- [CPList](#)  
*The list of control points.*
- [CPData](#)  
*The Control point data array.*
- [Objects](#)  
*The object array.*
- [\\_ch\\_oldgrab](#)  
*Used save the old grab.*
- [\\_ch\\_oldfocus](#)  
*Used save the old focus.*

### 9.36.1 Detailed Description

Main CTC Panel megawidget.

This megawidget implements two display areas, each with a vertical (Y) scrollbar. They share a horizontal (X) scrollbar. The upper display area contains schematic trackwork and the lower display area contains various switches, buttons, and lamps that deal with trackage control points.

#### Parameters

<i>path</i>	Pathname of the widget.
...	Options: <ul style="list-style-type: none"><li>• <code>-schematicbackground</code> The background color of the schematic display. Defaults to black.</li><li>• <code>-controlbackground</code> The background color of the control display. Defaults to darkgreen.</li><li>• <code>-width</code> The total width of the megawidget.</li><li>• <code>-height</code> The total height of the megawidget.</li></ul>

#### Author

Robert Heller <heller@deepsoft.com>

### 9.36.2 Constructor & Destructor Documentation

#### 9.36.2.1 CTCPanel()

```
CTCPanel::CTCPanel::CTCPanel (
    name ,
    ... )
```

Build and install all component widgets and process configuration.

#### Parameters

...	Argument list (option value pairs). Gets passed to the implicitly defined <code>configurelist</code> method.
-----	--

### 9.36.3 Member Function Documentation

#### 9.36.3.1 `_crosshairEnd()`

```
CTCPanel::CTCPanel::_crosshairEnd (
    canvas ,
    xvar ,
    yvar ,
    mx ,
    my ) [private]
```

Bound to left button down.

Ends crosshairs and returns the results.

##### Parameters

<i>canvas</i>	The canvas the crosshairs are on.
<i>xvar</i>	The name of the global variable to receive the X result.
<i>yvar</i>	The name of the global variable to receive the Y result.
<i>mx</i>	The X mouse position.
<i>my</i>	The Y mouse position.

#### 9.36.3.2 `_crosshairMove()`

```
CTCPanel::CTCPanel::_crosshairMove (
    canvas ,
    xvar ,
    yvar ,
    mx ,
    my ) [private]
```

Bound to mouse movement events.

##### Parameters

<i>canvas</i>	The canvas the crosshairs are on.
<i>xvar</i>	The name of the global variable to receive the X result.
<i>yvar</i>	The name of the global variable to receive the Y result.
<i>mx</i>	The X mouse position.
<i>my</i>	The Y mouse position.

### 9.36.3.3 \_crosshairStart()

```
CTCPanel::CTCPanel::_crosshairStart (
    canvas ,
    xvar ,
    yvar ) [private]
```

Start using the crosshairs.

#### Parameters

<i>canvas</i>	The canvas the crosshairs are on.
<i>xvar</i>	The name of the global variable to receive the X result.
<i>yvar</i>	The name of the global variable to receive the Y result.

### 9.36.3.4 \_CtcMainHScroll2()

```
CTCPanel::CTCPanel::_CtcMainHScroll2 (
    ... ) [private]
```

Internal method to scroll two canvases at the same time.

Bound to the horizontal scrollbar's -command.

#### Parameters

...	The arguments passed from the scroll bar.
-----	---

### 9.36.3.5 \_CtcMainSyncX()

```
CTCPanel::CTCPanel::_CtcMainSyncX (
    this ,
    other ,
    first ,
    last ) [private]
```

Internal method to x scroll updates.

Updates the scrolling for both canvases, making sure that they are in sync. The scrollbar is also updated. This method is bound to the -xscrollcommands of the schematic and controls canvases.

## Parameters

<i>this</i>	The canvas whose scrolling changed.
<i>other</i>	The other canvas, which needs to be synchronized.
<i>first</i>	The coordinate of the first (left most) visible part of the canvas. Passed from the canvas.
<i>last</i>	The coordinate of the last (right most) visible part of the canvas. Passed from the canvas.

9.36.3.6 `_PosInteger()`

```
CTCPanel::CTCPanel::_PosInteger (
    option ,
    value ) [private]
```

Method to validate a positive non zero integer option.

## Parameters

<i>option</i>	The option name.
<i>value</i>	The value to validate.

9.36.3.7 `bind()`

```
CTCPanel::CTCPanel::bind (
    name ,
    ... )
```

Method to set a binding on an Object.

## Parameters

<i>name</i>	The name of the object to set a binding on.
<i>sequence</i>	The event sequence to bind to.
<i>script</i>	The script to run when the binding event occurs. If the script is prefixed with a "+", it is appended to any existing script.

## Returns

The empty string in all cases where script is non-empty. If the script is missing, returns the current binding for the specified sequence. If neither sequence nor script is supplied, then returns a list of all bindings. See the `bind` sub-command of `canvas`.

9.36.3.8 checkInitCP()

```
CTCPanel::CTCPanel::checkInitCP (
    cp ) [protected]
```

Method to check that a control point has been initialized.

Should only be called from object constructors.

Parameters

<i>cp</i>	The name of the control point.
-----------	--------------------------------

9.36.3.9 controls\_crosshair()

```
CTCPanel::CTCPanel::controls_crosshair (
    ... )
```

Display crosshairs on the controls canvas.

Parameters

...	Options: <ul style="list-style-type: none"><li>-xvar Global variable to receive the X result.</li><li>-yvar Global variable to receive the Y result.</li></ul>
-----	--

9.36.3.10 coords()

```
CTCPanel::CTCPanel::coords (
    name ,
    tname )
```

Method to fetch the coordinates of some part of an object.

Parameters

<i>name</i>	The name of the object to fetch coordinates from.
<i>tname</i>	The name of the terminal of the object to fetch the coordinates of. See the individual element descriptions for valid terminal names.

Generated by Doxygen

#### 9.36.3.11 cplist()

```
CTCPanel::CTCPanel::cplist ( )
```

Method to return the list of controlpoints.

#### 9.36.3.12 create\_CodeButton()

```
CTCPanel::CTCPanel::create_CodeButton (
    name ,
    ... )
```

Method to create a code button object.

##### Parameters

<i>name</i>	The name of the new code button.
...	The argument list for the object constructor.

See [CodeButton](#) for details.

#### 9.36.3.13 create\_Crossing()

```
CTCPanel::CTCPanel::create_Crossing (
    name ,
    ... )
```

Method to create a track crossing object.

##### Parameters

<i>name</i>	The name of the new crossing.
...	The argument list for the object constructor.

See [Crossing](#) for details.

#### 9.36.3.14 create\_Crossover()

```
CTCPanel::CTCPanel::create_Crossover (
    name ,
    ... )
```



Method to create a crossover object.

#### Parameters

<i>name</i>	The name of the new crossover.
...	The argument list for the object constructor.

See [Crossover](#) for details.

#### 9.36.3.15 create\_CTCLabel()

```
CTCPanel::CTCPanel::create_CTCLabel (
    name ,
    ... )
```

Method to create a CTC Label label object.

#### Parameters

<i>name</i>	The name of the new label.
...	The argument list for the object constructor.

See [CTCLabel](#) for details.

#### 9.36.3.16 create\_CurvedBlock()

```
CTCPanel::CTCPanel::create_CurvedBlock (
    name ,
    ... )
```

Method to create a curved block of track object.

#### Parameters

<i>name</i>	The name of the new track block.
...	The argument list for the object constructor.

See [CurvedBlock](#) for details.

#### 9.36.3.17 create\_DoubleSlip()

```
CTCPanel::CTCPanel::create_DoubleSlip (
    name ,
    ... )
```

Method to create a double slip object.

## Parameters

<i>name</i>	The name of the new switch.
...	The argument list for the object constructor.

See [DoubleSlip](#) for details.

**9.36.3.18 create\_EndBumper()**

```
CTCPanel::CTCPanel::create_EndBumper (
    name ,
    ... )
```

Method to create a end bumper of track object.

## Parameters

<i>name</i>	The name of the new track block.
...	The argument list for the object constructor.

See [EndBumper](#) for details.

**9.36.3.19 create\_HiddenBlock()**

```
CTCPanel::CTCPanel::create_HiddenBlock (
    name ,
    ... )
```

Method to create a hidden block of track object.

## Parameters

<i>name</i>	The name of the new track block.
...	The argument list for the object constructor.

See [HiddenBlock](#) for details.

**9.36.3.20 create\_Lamp()**

```
CTCPanel::CTCPanel::create_Lamp (
    name ,
    ... )
```

Method to create a lamp object.

## Parameters

<i>name</i>	The name of the new lamp.
...	The argument list for the object constructor.

See [Lamp](#) for details.

## 9.36.3.21 create\_PushButton()

```
CTCPanel::CTCPanel::create_PushButton (
    name ,
    ... )
```

Method to create a push button object.

## Parameters

<i>name</i>	The name of the push button.
...	The argument list for the object constructor.

See [PushButton](#) for details.

## 9.36.3.22 create\_SchLabel()

```
CTCPanel::CTCPanel::create_SchLabel (
    name ,
    ... )
```

Method to create a schematic label object.

## Parameters

<i>name</i>	The name of the new label.
...	The argument list for the object constructor.

See [SchLabel](#) for details.

## 9.36.3.23 create\_ScissorCrossover()

```
CTCPanel::CTCPanel::create_ScissorCrossover (
    name ,
    ... )
```

Method to create a scissor crossover object.

**Parameters**

<i>name</i>	The name of the new crossover.
...	The argument list for the object constructor.

See [ScissorCrossover](#) for details.

**9.36.3.24 create\_Signal()**

```
CTCPanel::CTCPanel::create_Signal (
    name ,
    ... )
```

Method to create a signal object.

**Parameters**

<i>name</i>	The name of the signal.
...	The argument list for the object constructor.

See [Signal](#) for details.

**9.36.3.25 create\_SIGPlate()**

```
CTCPanel::CTCPanel::create_SIGPlate (
    name ,
    ... )
```

Method to create a signal plate object.

**Parameters**

<i>name</i>	The name of the new signal plate.
...	The argument list for the object constructor.

See [SIGPlate](#) for details.

**9.36.3.26 create\_SingleSlip()**

```
CTCPanel::CTCPanel::create_SingleSlip (
    name ,
    ... )
```

Method to create a single slip object.

## Parameters

<i>name</i>	The name of the new switch.
...	The argument list for the object constructor.

See [SingleSlip](#) for details.

**9.36.3.27 create\_StraightBlock()**

```
CTCPanel::CTCPanel::create_StraightBlock (
    name ,
    ... )
```

Method to create a straight block of track object.

## Parameters

<i>name</i>	The name of the new track block.
...	The argument list for the object constructor.

See [StraightBlock](#) for details.

**9.36.3.28 create\_StubYard()**

```
CTCPanel::CTCPanel::create_StubYard (
    name ,
    ... )
```

Method to create a stub (deadend) yard object.

## Parameters

<i>name</i>	The name of the new yard.
...	The argument list for the object constructor.

See [StubYard](#) for details.

**9.36.3.29 create\_Switch()**

```
CTCPanel::CTCPanel::create_Switch (
    name ,
    ... )
```

Method to create a switch (turnout) object.

## Parameters

<i>name</i>	The name of the new switch.
...	The argument list for the object constructor.

See [Switch](#) for details.

**9.36.3.30 create\_SWPlate()**

```
CTCPanel::CTCPanel::create_SWPlate (
    name ,
    ... )
```

Method to create a switch plate object.

## Parameters

<i>name</i>	The name of the new switch plate.
...	The argument list for the object constructor.

See [SWPlate](#) for details.

**9.36.3.31 create\_ThreeWaySW()**

```
CTCPanel::CTCPanel::create_ThreeWaySW (
    name ,
    ... )
```

Method to create a three way switch object.

## Parameters

<i>name</i>	The name of the new switch.
...	The argument list for the object constructor.

See [ThreeWaySW](#) for details.

**9.36.3.32 create\_ThroughYard()**

```
CTCPanel::CTCPanel::create_ThroughYard (
    name ,
    ... )
```

Method to create a through yard object.

## Parameters

<i>name</i>	The name of the new yard.
...	The argument list for the object constructor.

See [ThroughYard](#) for details.

**9.36.3.33 create\_Toggle()**

```
CTCPanel::CTCPanel::create_Toggle (
    name ,
    ... )
```

Method to create a toggle switch object.

## Parameters

<i>name</i>	The name of the new toggle switch.
...	The argument list for the object constructor.

See [Toggle](#) for details.

**9.36.3.34 delete()**

```
CTCPanel::CTCPanel::delete (
    name )
```

Delete a named object.

## Parameters

<i>name</i>	The name of the object to delete.
-------------	-----------------------------------

**9.36.3.35 exists()**

```
CTCPanel::CTCPanel::exists (
    name )
```

Test if the named object exists.

**Parameters**

<i>name</i>	The object to test for.
-------------	-------------------------

**9.36.3.36   geti()**

```
CTCPanel::CTCPanel::geti (
    name ,
    ind )
```

Method to get the indicator state of an object.

**Parameters**

<i>name</i>	The name of the object to fetch the indicator state of.
<i>ind</i>	The indicator whose state is return. See the individual element descriptions for valid indicator names.

**9.36.3.37   getv()**

```
CTCPanel::CTCPanel::getv (
    name )
```

Method to get the value (or state) of an object.

**Parameters**

<i>name</i>	The name of the object to fetch the value of.
-------------	---

**9.36.3.38   getZoom()**

```
CTCPanel::CTCPanel::getZoom ( )
```

Return the zoom (scaling) factor.



#### 9.36.3.39 invoke()

```
CTCPanel::CTCPanel::invoke (
    name )
```

Method to invoke an object.

Returns true if the element is occupied.

##### Parameters

<i>name</i>	The name of the object to invoke.
-------------	-----------------------------------

See the individual object invoke methods for details.

#### 9.36.3.40 itemcget()

```
CTCPanel::CTCPanel::itemcget (
    name ,
    option )
```

Method to get a configuration option from an object.

##### Parameters

<i>name</i>	The object whose configuration option is to be fetched from.
<i>option</i>	The option to fetch. See the individual element descriptions for valid options.

#### 9.36.3.41 itemconfigure()

```
CTCPanel::CTCPanel::itemconfigure (
    name ,
    ... )
```

Method to set a configuration option from an object.

##### Parameters

<i>name</i>	The object whose configuration option is to be configured.
...	The configuration arguments.

#### 9.36.3.42 lappendCP()

```
CTCPanel::CTCPanel::lappendCP (
    cp ,
    slot ,
    what ) [protected]
```

Method to lappend something to a slot in a control point's data.

Should only be called from object constructors.

##### Parameters

<i>cp</i>	The control point to update.
<i>slot</i>	The slot to update.
<i>what</i>	The object to add to the slot.

#### 9.36.3.43 lremoveCP()

```
CTCPanel::CTCPanel::lremoveCP (
    cp ,
    slot ,
    what ) [protected]
```

Method to remove something from a slot in a control point's data.

Should only be called from object destructors.

##### Parameters

<i>cp</i>	The control point to update.
<i>slot</i>	The slot to update.
<i>what</i>	The object to remove from the slot.

#### 9.36.3.44 move()

```
CTCPanel::CTCPanel::move (
    name ,
    x ,
    y )
```

Move a named object.

## Parameters

<i>name</i>	The name of the object to be moved.
<i>x</i>	The amount of the x movement.
<i>y</i>	The amount of the y movement.

## 9.36.3.45 objectlist()

```
CTCPanel::CTCPanel::objectlist (
    cp = "",
    class = "" )
```

Method to return the list of objects.

## Parameters

<i>cp</i>	(optional) The name of control point to return the object of. If cp is the empty string, return all objects.
<i>class</i>	(optional) The class of objects to return. If class is the empty string, return all classes of objects.

## 9.36.3.46 print()

```
CTCPanel::CTCPanel::print (
    name ,
    fp )
```

Method to print the named object to the specified file channel.

## Parameters

<i>name</i>	The object to print.
<i>fp</i>	The file channel to print to.

## 9.36.3.47 schematic\_crosshair()

```
CTCPanel::CTCPanel::schematic_crosshair (
    ... )
```

Display crosshairs on the schematic canvas.

## Parameters

...	Options:
	<ul style="list-style-type: none"> <li>• -xvar Global variable to receive the X result.</li> <li>• -yvar Global variable to receive the Y result.</li> </ul>

## 9.36.3.48 seti()

```
CTCPanel::CTCPanel::seti (
    name ,
    ind ,
    value )
```

Method to set the indicator state of an object.

## Parameters

<i>name</i>	The name of the object whose indicator state is to be set.
<i>ind</i>	The indicator to update. See the individual element descriptions for valid indicator names.
<i>value</i>	The new indicator value, generally on or off.

## 9.36.3.49 setv()

```
CTCPanel::CTCPanel::setv (
    name ,
    value )
```

Method to set the value (or state) of an object.

## Parameters

<i>name</i>	The name of the object to update.
<i>value</i>	The value to set it to. See the individual element descriptions for valid values.

#### 9.36.3.50 setZoom()

```
CTCPanel::CTCPanel::setZoom (
    zoomFactor )
```

Method to set the zoom factor to a specific factor.

##### Parameters

<i>zoomFactor</i>	The zoom factor.
-------------------	------------------

#### 9.36.3.51 updateAndSyncCP()

```
CTCPanel::CTCPanel::updateAndSyncCP (
    cp ) [protected]
```

Method to update and synchronize a control point.

Should only be called from object methods.

##### Parameters

<i>cp</i>	The name of the control point.
-----------	--------------------------------

#### 9.36.3.52 updateSR()

```
CTCPanel::CTCPanel::updateSR (
    canvas ,
    newheight ,
    newwidth )
```

Method to update one of the canvases scroll region.

Bound to the Configure event of each of the canvases.

##### Parameters

<i>canvas</i>	The canvas to update.
<i>newheight</i>	The new height.
<i>newwidth</i>	The new width.

#### 9.36.3.53 zoomBy()

```
CTCPanel::CTCPanel::zoomBy (  
    zoomFactor )
```

Method to zoom the display by a zoom factor.

##### Parameters

<i>zoomFactor</i>	The zoom factor.
-------------------	------------------

### 9.36.4 Member Data Documentation

#### 9.36.4.1 \_ch\_oldfocus

```
CTCPanel::CTCPanel::_ch_oldfocus [private]
```

Used save the old focus.

#### 9.36.4.2 \_ch\_oldgrab

```
CTCPanel::CTCPanel::_ch_oldgrab [private]
```

Used save the old grab.

#### 9.36.4.3 controls

```
CTCPanel::CTCPanel::controls [private]
```

The controls component.

#### 9.36.4.4 controlsYscroll

CTCPanel::CTCPanel::controlsYscroll [private]

The controls y scroll bar component.

#### 9.36.4.5 CPData

CTCPanel::CTCPanel::CPData [private]

The Control point data array.

#### 9.36.4.6 CPList

CTCPanel::CTCPanel::CPList [private]

The list of control points.

#### 9.36.4.7 Objects

CTCPanel::CTCPanel::Objects [private]

The object array.

#### 9.36.4.8 scale

CTCPanel::CTCPanel::scale [private]

The current scale value.

#### 9.36.4.9 schematic

CTCPanel::CTCPanel::schematic [private]

The schematic component.

#### 9.36.4.10 schematicYscroll

CTCPanel::CTCPanel::schematicYscroll [private]

The schematic y scroll bar component.

#### 9.36.4.11 xscroll

CTCPanel::CTCPanel::xscroll [private]

The shared x scroll bar component.

## 9.37 ctiacela::CTIAcela Class Reference

Main [CTIAcela](#) interface class.

### Public Member Functions

- [CTIAcela](#) (name, port,...)  
*Constructor: open a connection to the CTI Acela.*
- [~CTIAcela](#) ()  
*The destructor restores the serial port's state and closes it.*
- [HaveData](#) ()
- [OnlineP](#) ()
- [Activate](#) (address)  
*Activate a control.*
- [Deactive](#) (address)  
*Deactive a control.*
- [PulseOn](#) (address, pulsewidth)  
*Pulse On a control.*
- [PulseOff](#) (address, pulsewidth)  
*Pulse Off a control.*
- [Blink](#) (address, pulsewidth)  
*Blink a control.*
- [ReverseBlink](#) (address, pulsewidth)  
*Reverse Blink a control.*
- [Control4](#) (address, c1, c2, c3, c4)  
*Configure 4 controls simultaneously.*
- [Control8](#) (address, c1, c2, c3, c4, c5, c6, c7, c8)  
*Configure 8 controls simultaneously.*
- [Control16](#) (address, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12, c13, c14, c15, c16)  
*Configure 16 controls simultaneously.*



- [Throttle](#) (address, speed, momentum, brake, direction, idle)  
*Throttle command.*
- [EmergencyStop](#) ()  
*Emergency Stop.*
- [Signal2](#) (address, lamp1, lamp2, yellow="off")  
*Control 2-lamp signals.*
- [Signal3](#) (address, lamp1, lamp2, lamp3)  
*Control 3-lamp signals.*
- [Signal4](#) (address, lamp1, lamp2, lamp3, lamp4)  
*Control 4-lamp signals.*
- [SignalSettings](#) (blinkrate, yellowhue)  
*Set Signal Settings.*
- [SignalBrightness](#) (brightness)  
*Set signal brightness.*
- [ConfigureSensor](#) (address, threshold, select, polarity)  
*Configure a sensor.*
- [Read](#) (address)  
*Read the state of a sensor.*
- [Read4](#) (address)  
*Read the state of four sensors.*
- [Read8](#) (address)  
*Read the state of eight sensors.*
- [Read16](#) (address)  
*Read the state of sixteen sensors.*
- [ReadAll](#) ()  
*Read all sensors.*
- [SRQControl](#) (enable="yes")  
*Enable or disable SRQ messages.*
- [Query](#) ()  
*Query sensor change state.*
- [ResetNetwork](#) ()  
*Reset the network.*
- [NetworkOnline](#) ()  
*Bring the network online.*
- [NetworkOffline](#) ()  
*Bring the network offline.*
- [Poll](#) ()  
*Poll the network configuration.*
- [ReadRevision](#) ()  
*Read CTI Acela firmware revision.*

## Static Public Member Functions

- static [validate](#) (object)  
*Type validation method.*

## Private Member Functions

- [\\_handleSRQ](#) ()  
*Handle a service request.*
- [\\_transmit](#) (buffer, responsebytes=0)  
*Transmit buffer and wait for response.*
- [\\_readevent](#) ()  
*Read event method.*
- [\\_readbyte](#) (thebytevar)  
*Read a single byte from the serial interface.*

## Static Private Member Functions

- static [highbyte](#) (addr)  
*Return the high byte of address.*
- static [lowbyte](#) (addr)  
*Return the low byte of address.*
- static [pack4](#) (b1, b2, b3, b4)  
*pack 4 bits*
- static [pack8](#) (b1, b2, b3, b4, b5, b6, b7, b8)  
*pack 4 bits*

## Private Attributes

- [ttyfd](#)  
*Terminal file descriptor.*
- [dataavailable](#)  
*Flag set to true (yes) when sensor data is available.*
- [networkonline](#)  
*Flag set to false (no) when the network goes offline.*
- [\\_timeout](#)  
*Timeout flag.*

## Static Private Attributes

- static [Responses](#)  
*Responses.*
- static [Opcodes](#)  
*Opcodes.*
- static [LampBits](#)  
*Lamp Bits.*
- static [FilterSelectBits](#)  
*Filter Select Bits.*
- static [CTI\\_DeviceMap](#)  
*CTI Module Map.*
- static [maxtries](#)  
*Loop control for read attempts.*

### 9.37.1 Detailed Description

Main [CTIAcela](#) interface class.

## Parameters

<i>name</i>	Name of the <a href="#">CTIAcela</a> interface instance.
<i>port</i>	Name of the serial port connected to the CTI Acela. Either something like /dev/ttySN for real serial ports or /dev/ttyACM0 for a USB connected Acela.

## Author

Robert Heller <heller@deepsoft.com>

## 9.38 CTCPanel::CurvedBlock Class Reference

Curved Block object type.

### Public Member Functions

- [CurvedBlock](#) (name, \_ctcpanel, \_canvas,...)  
*Construct a [CurvedBlock](#) object.*
- [~CurvedBlock](#) ()  
*Clean up all data objects and free up all resources.*
- [setv](#) (value)  
*Method to set out value (state).*
- [geti](#) (ind)  
*Method to get the state of one of our indicators (none).*
- [seti](#) (ind, value)  
*Method to set an indicator's state (none).*
- [invoke](#) ()  
*Method to invoke the [CurvedBlock](#).*

### Private Member Functions

- [\\_configureLabel](#) (option, value)  
*Method to update the label option.*

### Static Private Member Functions

- static [\\_square](#) (x)  
*Typemethod to compute the square of a number.*
- static [\\_RadiansToDegrees](#) (rads)  
*Typemethod to convert from radians to degrees.*

## Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*

## Static Private Attributes

- static [\\_PI](#)  
*PI is  $\pi$ .*

### 9.38.1 Detailed Description

Curved Block object type.

These are on the schematic and represent a piece of track on the Schematic.

#### Parameters

<a href="#">_ctcpanel</a>	The <a href="#">CTCPanel</a> megawidget.
<a href="#">_canvas</a>	The schematic canvas to draw the <a href="#">CurvedBlock</a> on.
...	Options: <ul style="list-style-type: none"> <li>• -x1 The first x coordinate of the object (readonly, default 0).</li> <li>• -y1 The first y coordinate of the object (readonly, default 0).</li> <li>• -x2 The second x coordinate of the object (readonly, default 0).</li> <li>• -y2 The second y coordinate of the object (readonly, default 0).</li> <li>• -radius The radius of the curve (readonly, default 10).</li> <li>• -controlpoint The name of the control point this label is part of (readonly, default MainLine).</li> <li>• -label The label of the <a href="#">CurvedBlock</a> (default "").</li> <li>• -position The position of the label (readonly, default below).</li> <li>• -occupiedcommand A command to run to find out if the block is occupied (default {}).</li> </ul>

Defined coords terminals:

- E1 First endpoint.
- E2 Second endpoint.

Defined values (states): none. Defined indicators: none.

## Author

Robert Heller <heller@deepsoft.com>

## 9.38.2 Constructor & Destructor Documentation

### 9.38.2.1 CurvedBlock()

```
CTCPanel::CurvedBlock::CurvedBlock (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a [CurvedBlock](#) object.

See @ for an explanation of the underlying math.

## Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the <a href="#">CurvedBlock</a> on.
<code>...</code>	Option list.

### 9.38.2.2 ~CurvedBlock()

```
CTCPanel::CurvedBlock::~~CurvedBlock ( )
```

Clean up all data objects and free up all resources.

## 9.38.3 Member Function Documentation

### 9.38.3.1 \_configureLabel()

```
CTCPanel::CurvedBlock::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

### 9.38.3.2 \_RadiansToDegrees()

```
static CTCPanel::CurvedBlock::_RadiansToDegrees (
    rads ) [static], [private]
```

Type method to convert from radians to degrees.

### 9.38.3.3 \_square()

```
static CTCPanel::CurvedBlock::_square (
    x ) [static], [private]
```

Type method to compute the square of a number.

### 9.38.3.4 geti()

```
CTCPanel::CurvedBlock::geti (
    ind )
```

Method to get the state of one of our indicators (none).

### 9.38.3.5 invoke()

```
CTCPanel::CurvedBlock::invoke ( )
```

Method to invoke the [CurvedBlock](#).

### 9.38.3.6 seti()

```
CTCPanel::CurvedBlock::seti (
    ind ,
    value )
```

Method to set an indicator's state (none).

### 9.38.3.7 setv()

```
CTCPanel::CurvedBlock::setv (
    value )
```

Method to set out value (state).

## Parameters

<i>value</i>	The new state to set.
--------------	-----------------------

## 9.38.4 Member Data Documentation

9.38.4.1 `_PI`

`CTCPanel::CurvedBlock::_PI` [static], [private]

PI is  $\pi$ .

9.38.4.2 `canvas`

`CTCPanel::CurvedBlock::canvas` [private]

The canvas component (parent widget component).

9.38.4.3 `ctcpanel`

`CTCPanel::CurvedBlock::ctcpanel` [private]

The CTC Panel component (parent widget).

## 9.39 Instruments::DialInstrument Class Reference

Generic dial instrument.

## Public Member Functions

- [DialInstrument](#) (name, \_canvas,...)  
*Constructor – initialize a [DialInstrument](#).*
- [~DialInstrument](#) ()  
*Destructor – free up all resources.*
- [setvalue](#) (value, value2=0)  
*Method to set the value of the dial instrument.*



### Private Attributes

- [ValueRange](#)  
*Value range.*
- [dTextX](#)  
*X position of the dial text.*
- [dTextY](#)  
*Y position of the dial text.*

#### 9.39.1 Detailed Description

Generic dial instrument.

##### Parameters

<code>_canvas</code>	The canvas to draw the dial instrument on.
----------------------	--

## Parameters

...	<p>Options:</p> <ul style="list-style-type: none"> <li>• -x The X coordinate of the instrument (default 0).</li> <li>• -y The Y coordinate of the instrument (default 0).</li> <li>• -size The size of the instrument (default 100).</li> <li>• -label The label of the instrument (default <a href="#">DialInstrument</a>).</li> <li>• -labelcolor The color of the label (default black).</li> <li>• -labelfont The font of the label (default {Times 14 bold}).</li> <li>• -background The background color of the instrument (default blue).</li> <li>• -outline The outline color of the instrument (default black).</li> <li>• -scaleback The background color of the scale (default white).</li> <li>• -scaleticks The color of the scale ticks (default black).</li> <li>• -fontfamily The font family used on the instrument (default Courier).</li> <li>• -maxvalue The maximum value (default 100).</li> <li>• -minvalue The minimum value (default 0).</li> <li>• -minat The pointer position, in degrees, of the minimum value (default 225).</li> <li>• -maxat The pointer position, in degrees, of the maximum value (default 315).</li> <li>• -pointercolor The color of the pointer (default black).</li> <li>• -secondpointerp Should a second pointer be drawn (default no)?</li> <li>• -secondpointercolor The color of the second pointer (default red).</li> <li>• -scaleticksinterval The interval of the scale ticks (default 10).</li> <li>• -digitalp Should a digital display also be included (default yes)?</li> <li>• -digits How many digits for the digital display (default 3).</li> <li>• -digitalbackground The background color of the digital display (default white).</li> <li>• -digitaldigitcolor The foreground color of the digital display (default black).</li> </ul>
-----	---

## Author

Robert Heller <heller@deepsoft.com>

## 9.39.2 Constructor &amp; Destructor Documentation

### 9.39.2.1 DialInstrument()

```
Instruments::DialInstrument::DialInstrument (
    name ,
    _canvas ,
    ... )
```

Constructor – initialize a [DialInstrument](#).

#### Parameters

<code>_canvas</code>	The canvas to draw the <a href="#">DialInstrument</a> on.
<code>...</code>	Option list.

### 9.39.2.2 ~DialInstrument()

```
Instruments::DialInstrument::~~DialInstrument ( )
```

Destructor – free up all resources.

## 9.39.3 Member Function Documentation

### 9.39.3.1 setvalue()

```
Instruments::DialInstrument::setvalue (
    value ,
    value2 = 0 )
```

Method to set the value of the dial instrument.

#### Parameters

<i>value</i>	The value to set the instrument to.
<i>value2</i>	The value for the second pointer.

## 9.39.4 Member Data Documentation



## Protected Member Functions

- virtual ostream & [WriteDictionaryElements](#) (ostream &stream) const  
*Write the elements of a dictionary.*

### 9.40.1 Detailed Description

PDF [Dictionary](#) class.

This base class is useless by itself. Real specific dictionaries will be derived from this class.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.40.2 Constructor & Destructor Documentation

#### 9.40.2.1 Dictionary()

```
FCFSupport::PDFFileStructures::Dictionary::Dictionary (
    unsigned long int objNum = 0L,
    unsigned short int genNum = 0,
    CrossReferenceTable * tab = NULL ) [inline]
```

Constructor.

Create a new dictionary.

#### Parameters

<i>objNum</i>	The next object number.
<i>genNum</i>	The generation number.
<i>tab</i>	The cross reference table we are in.

#### 9.40.2.2 ~Dictionary()

```
virtual FCFSupport::PDFFileStructures::Dictionary::~~Dictionary ( ) [inline], [virtual]
```

Destructor.

Clean everything up.

### 9.40.3 Member Function Documentation

#### 9.40.3.1 WriteDictionaryElements()

```
virtual ostream& FCFSupport::PDFFileStructures::Dictionary::WriteDictionaryElements (
    ostream & stream ) const [inline], [protected], [virtual]
```

Write the elements of a dictionary.

##### Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

Reimplemented in [FCFSupport::PDFFileStructures::InformationDirectory](#), [FCFSupport::PDFFileStructures::CatalogDictionary](#), [FCFSupport::PDFFileStructures::Type1FontDictionary](#), [FCFSupport::PDFFileStructures::FontDictionary](#), [FCFSupport::PDFFileStructures::PageLabelTree](#), [FCFSupport::PDFFileStructures::PageLabelDictionary](#), [FCFSupport::PDFFileStructures::PageTree](#), [FCFSupport::PDFFileStructures::Page](#), [FCFSupport::PDFFileStructures::ResourceDictionary](#), [FCFSupport::PDFFileStructures::IndirectObjectDictionary](#), and [FCFSupport::PDFFileStructures::TypedDictionary](#).

References [lcc::stream](#).

#### 9.40.3.2 WriteDirect()

```
virtual ostream& FCFSupport::PDFFileStructures::Dictionary::WriteDirect (
    ostream & stream ) const [inline], [virtual]
```

Write an object directly.

##### Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

Implements [FCFSupport::PDFFileStructures::IndirectObject](#).

References [lcc::stream](#).

## 9.41 Instruments::DigitalClock Class Reference

Digital clock instrument.

## Public Member Functions

- [DigitalClock](#) (name, \_canvas,...)  
*Constructor – initialize a digital clock.*
- [~DigitalClock](#) ()  
*Destructor – free up all resources.*
- [settime](#) (hour, minute)  
*Method to set the time.*

### 9.41.1 Detailed Description

Digital clock instrument.

#### Parameters

<code>_canvas</code>	The canvas to draw the digital clock instrument on.
<code>...</code>	Options: <ul style="list-style-type: none"> <li>• <code>-x</code> The X coordinate of the instrument (default 0).</li> <li>• <code>-y</code> The Y coordinate of the instrument (default 0).</li> <li>• <code>-size</code> The size of the instrument (default 100).</li> <li>• <code>-label</code> The label of the instrument (default Clock).</li> <li>• <code>-labelcolor</code> The color of the label (default black).</li> <li>• <code>-labelfont</code> The font of the label (default {Times 14 bold}).</li> <li>• <code>-background</code> The background color of the instrument (default blue).</li> <li>• <code>-outline</code> The outline color of the instrument (default black).</li> <li>• <code>-scaleback</code> The background color of the scale (default white).</li> <li>• <code>-scaleticks</code> The color of the scale ticks (default black).</li> <li>• <code>-fontfamily</code> The font family used on the instrument (default Courier).</li> <li>• <code>-digitcolor</code> The color of the digits (default black).</li> </ul>

#### Author

Robert Heller <heller@deepsoft.com>

### 9.41.2 Constructor & Destructor Documentation

### 9.41.2.1 DigitalClock()

```
Instruments::DigitalClock::DigitalClock (
    name ,
    _canvas ,
    ... )
```

Constructor – initialize a digital clock.

#### Parameters

<code>_canvas</code>	The canvas to draw the <a href="#">DialInstrument</a> on.
<code>...</code>	Option list.

### 9.41.2.2 ~DigitalClock()

```
Instruments::DigitalClock::~~DigitalClock ( )
```

Destructor – free up all resources.

## 9.41.3 Member Function Documentation

### 9.41.3.1 settime()

```
Instruments::DigitalClock::settime (
    hour ,
    minute )
```

Method to set the time.

#### Parameters

<i>hour</i>	The hour, an integer between 1 and 12.
<i>minute</i>	The minute, an integer between 0 and 59.

## 9.42 Instruments::DigitalInstrument Class Reference

Digital instrument.



## Public Member Functions

- [DigitalInstrument](#) (name, \_canvas,...)  
*Constructor – initialize a digital instrument.*
- [~DigitalInstrument](#) ()  
*Destructor – free up all resources.*
- [setvalue](#) (value)  
*Method to set the value of the digital instrument.*

### 9.42.1 Detailed Description

Digital instrument.

#### Parameters

<code>_canvas</code>	The canvas to draw the digital instrument on.
<code>...</code>	Options: <ul style="list-style-type: none"> <li>• <code>-x</code> The X coordinate of the instrument (default 0).</li> <li>• <code>-y</code> The Y coordinate of the instrument (default 0).</li> <li>• <code>-size</code> The size of the instrument (default 100).</li> <li>• <code>-label</code> The label of the instrument (default <a href="#">DigitalInstrument</a>).</li> <li>• <code>-labelcolor</code> The color of the label (default black).</li> <li>• <code>-labelfont</code> The font of the label (default {Times 14 bold}).</li> <li>• <code>-background</code> The background color of the instrument (default blue).</li> <li>• <code>-outline</code> The outline color of the instrument (default black).</li> <li>• <code>-scaleback</code> The background color of the scale (default white).</li> <li>• <code>-scaleticks</code> The color of the scale ticks (default black).</li> <li>• <code>-fontfamily</code> The font family used on the instrument (default Courier).</li> <li>• <code>-digitcolor</code> The color of the digits (default black).</li> <li>• <code>-digits</code> The number of digits (default 3).</li> </ul>

#### Author

Robert Heller <heller@deepsoft.com>

## 9.42.2 Constructor & Destructor Documentation

### 9.42.2.1 DigitalInstrument()

```
Instruments::DigitalInstrument::DigitalInstrument (
    name ,
    _canvas ,
    ... )
```

Constructor – initialize a digital instrument.

#### Parameters

<code>_canvas</code>	The canvas to draw the <a href="#">DialInstrument</a> on.
<code>...</code>	Option list.

### 9.42.2.2 ~DigitalInstrument()

```
Instruments::DigitalInstrument::~~DigitalInstrument ( )
```

Destructor – free up all resources.

## 9.42.3 Member Function Documentation

### 9.42.3.1 setvalue()

```
Instruments::DigitalInstrument::setvalue (
    value )
```

Method to set the value of the digital instrument.

#### Parameters

<code>value</code>	The value to set the instrument to.
--------------------	-------------------------------------

## 9.43 FCFSupport::Division Class Reference

The [Division](#) class implements a single division, which contains a number of contiguous stations.

```
#include <Division.h>
```

### Public Member Functions

- [Division](#) ()  
*Default constructor.*
- [Division](#) ([Division](#) &other)  
*Copy constructor.*
- [Division](#) & [operator=](#) ([Division](#) &other)  
*Assignment operator.*
- [Division](#) (char s, [FCFSupport::Industry](#) \*h, char a, const char \*n)  
*Constructor given a set of field values.*
- [~Division](#) ()  
*Destructor.*
- const char \* [Name](#) () const  
*Return the division's name.*
- [FCFSupport::Industry](#) \* [Home](#) () const  
*Return the division's home yard.*
- char [Symbol](#) () const  
*Return the division's Symbol.*
- char [Area](#) () const  
*Return the division's area.*
- int [NumberOfStations](#) () const  
*Return the number of stations in this division.*
- const [FCFSupport::Station](#) \* [TheStation](#) (int i) const  
*Return a selected station in the division.*
- void [AppendStation](#) ([FCFSupport::Station](#) \*station)  
*Append an additional station to this division.*

### Private Attributes

- string [name](#)  
*The name of the division.*
- [StationVector](#) [stations](#)  
*The vector of stations in the division.*
- [Industry](#) \* [home](#)  
*The division's home yard.*
- char [symbol](#)  
*The division's symbol.*
- char [area](#)  
*The division's area.*

## Friends

- class [System](#)

*The [System](#) class is a friend.*

### 9.43.1 Detailed Description

The [Division](#) class implements a single division, which contains a number of contiguous stations.

A division has a name, a symbol, an area, a home yard, and a list of stations.

```
@author Robert Heller \<heller\@deepsoft.com\>
```

### 9.43.2 Constructor & Destructor Documentation

#### 9.43.2.1 [Division\(\)](#) [1/3]

```
FCFSupport::Division::Division ( ) [inline]
```

Default constructor.

All fields are initialized to empty or NULL values.

References area, home, name, and symbol.

#### 9.43.2.2 [Division\(\)](#) [2/3]

```
FCFSupport::Division::Division (
    Division & other ) [inline]
```

Copy constructor.

A new division is created as a copy of an existing division.

#### Parameters

<i>other</i>	The other division.
--------------	---------------------

References area, home, name, stations, and symbol.

### 9.43.2.3 Division() [3/3]

```
FCFSupport::Division::Division (
    char s,
    FCFSupport::Industry * h,
    char a,
    const char * n ) [inline]
```

Constructor given a set of field values.

#### Parameters

<i>s</i>	The division's symbol.
<i>h</i>	The division's home yard.
<i>a</i>	The division's area.
<i>n</i>	The division's name.

References area, home, name, and symbol.

### 9.43.2.4 ~Division()

```
FCFSupport::Division::~~Division ( ) [inline]
```

Destructor.

## 9.43.3 Member Function Documentation

### 9.43.3.1 AppendStation()

```
void FCFSupport::Division::AppendStation (
    FCFSupport::Station * station ) [inline]
```

Append an additional station to this division.

#### Parameters

<i>station</i>	The station to append.
----------------	------------------------

References stations.

#### 9.43.3.2 Area()

```
char FCFSupport::Division::Area ( ) const [inline]
```

Return the division's area.

References area.

#### 9.43.3.3 Home()

```
FCFSupport::Industry* FCFSupport::Division::Home ( ) const [inline]
```

Return the division's home yard.

References home.

#### 9.43.3.4 Name()

```
const char* FCFSupport::Division::Name ( ) const [inline]
```

Return the division's name.

References name.

#### 9.43.3.5 NumberOfStations()

```
int FCFSupport::Division::NumberOfStations ( ) const [inline]
```

Return the number of stations in this division.

References stations.

#### 9.43.3.6 operator=()

```
Division& FCFSupport::Division::operator= (
    Division & other ) [inline]
```

Assignment operator.

Copy one division to another.

**Parameters**

<i>other</i>	The other division.
--------------	---------------------

References area, home, name, stations, and symbol.

**9.43.3.7 Symbol()**

```
char FCFSupport::Division::Symbol ( ) const [inline]
```

Return the division's Symbol.

References symbol.

**9.43.3.8 TheStation()**

```
const FCFSupport::Station* FCFSupport::Division::TheStation (
    int i ) const [inline]
```

Return a selected station in the division.

**Parameters**

<i>i</i>	The station index.
----------	--------------------

References stations.

**9.43.4 Friends And Related Function Documentation****9.43.4.1 System**

```
friend class System [friend]
```

The [System](#) class is a friend.

**9.43.5 Member Data Documentation**

#### 9.43.5.1 area

```
char FCFSupport::Division::area [private]
```

The division's area.

Referenced by Area(), Division(), and operator=().

#### 9.43.5.2 home

```
Industry* FCFSupport::Division::home [private]
```

The division's home yard.

Referenced by Division(), Home(), and operator=().

#### 9.43.5.3 name

```
string FCFSupport::Division::name [private]
```

The name of the division.

Referenced by Division(), Name(), and operator=().

#### 9.43.5.4 stations

```
StationVector FCFSupport::Division::stations [private]
```

The vector of stations in the division.

Referenced by AppendStation(), Division(), NumberOfStations(), operator=(), and TheStation().

#### 9.43.5.5 symbol

```
char FCFSupport::Division::symbol [private]
```

The division's symbol.

Referenced by Division(), operator=(), and Symbol().



## 9.44 xpressnet::DoubleHeaderInformation Class Reference

Double header information.

### Public Member Functions

- [Address](#) ()  
*Return address.*
- [Available](#) ()  
*Return available flag.*
- [Direction](#) ()  
*Return direction.*
- [SpeedStepMode](#) ()  
*Return speed step mode.*
- [Speed](#) ()  
*Return speed.*
- [Address2](#) ()  
*Return the address of second unit in double header.*
- [Function](#) (f)  
*Return function status.*
- [DoubleHeaderInformation](#) (name, a, addr2, avail, dir, ssm, s, f0, f1, f2, f3, f4, f5=0, f6=0, f7=0, f8=0, f9=0, f10=0, f11=0, f12=0)  
*Constructor.*

### Private Attributes

- [\\_address](#)  
*Locomotive address.*
- [\\_available](#)  
*Locomotive is available.*
- [\\_direction](#)  
*Locomotive direction.*
- [\\_speedstep](#)  
*Locomotive speed step mode.*
- [\\_speed](#)  
*Locomotive speed.*
- [\\_function0](#)  
*Function 0.*
- [\\_function1](#)  
*Function 1.*
- [\\_function2](#)  
*Function 2.*
- [\\_function3](#)  
*Function 3.*
- [\\_function4](#)

- Function 4.*
    - [\\_function5](#)
  - Function 5.*
    - [\\_function6](#)
  - Function 6.*
    - [\\_function7](#)
  - Function 7.*
    - [\\_function8](#)
  - Function 8.*
    - [\\_function9](#)
  - Function 9.*
    - [\\_function10](#)
  - Function 10.*
    - [\\_function11](#)
  - Function 11.*
    - [\\_function12](#)
  - Function 12.*
    - [\\_address2](#)
- Double header address.*

### 9.44.1 Detailed Description

Double header information.

Author

Robert Heller <heller@deepsoft.com>

### 9.44.2 Constructor & Destructor Documentation

#### 9.44.2.1 DoubleHeaderInformation()

```
xpressnet::DoubleHeaderInformation::DoubleHeaderInformation (
    name ,
    a ,
    addr2 ,
    avail ,
    dir ,
    ssm ,
    s ,
    f0 ,
    f1 ,
    f2 ,
    f3 ,
```

```

f4 ,
f5  = 0,
f6  = 0,
f7  = 0,
f8  = 0,
f9  = 0,
f10 = 0,
f11 = 0,
f12 = 0 )

```

Constructor.

#### Parameters

<i>a</i>	Locomotive address.
<i>addr2</i>	Double header address.
<i>avail</i>	Available flag.
<i>dir</i>	Direction.
<i>ssm</i>	Speed step mode.
<i>s</i>	Locomotive speed.
<i>f0</i>	Function 0 status.
<i>f1</i>	Function 1 status.
<i>f2</i>	Function 2 status.
<i>f3</i>	Function 3 status.
<i>f4</i>	Function 4 status.
<i>f5</i>	Function 5 status.
<i>f6</i>	Function 6 status.
<i>f7</i>	Function 7 status.
<i>f8</i>	Function 8 status.
<i>f9</i>	Function 9 status.
<i>f10</i>	Function 10 status.
<i>f11</i>	Function 11 status.
<i>f12</i>	Function 12 status.

### 9.44.3 Member Function Documentation

#### 9.44.3.1 Address()

```
xpressnet::DoubleHeaderInformation::Address ( )
```

Return address.

#### 9.44.3.2 Address2()

```
xpressnet::DoubleHeaderInformation::Address2 ( )
```

Return the address of second unit in double header.

#### 9.44.3.3 Available()

```
xpressnet::DoubleHeaderInformation::Available ( )
```

Return available flag.

#### 9.44.3.4 Direction()

```
xpressnet::DoubleHeaderInformation::Direction ( )
```

Return direction.

#### 9.44.3.5 Function()

```
xpressnet::DoubleHeaderInformation::Function (
    f )
```

Return function status.

##### Parameters

<i>f</i>	Function whose status to return.
----------	----------------------------------

#### 9.44.3.6 Speed()

```
xpressnet::DoubleHeaderInformation::Speed ( )
```

Return speed.

#### 9.44.3.7 SpeedStepMode()

```
xpressnet::DoubleHeaderInformation::SpeedStepMode ( )
```

Return speed step mode.

### 9.44.4 Member Data Documentation

#### 9.44.4.1 \_address

```
xpressnet::DoubleHeaderInformation::_address [private]
```

Locomotive address.

#### 9.44.4.2 \_address2

```
xpressnet::DoubleHeaderInformation::_address2 [private]
```

Double header address.

#### 9.44.4.3 \_available

```
xpressnet::DoubleHeaderInformation::_available [private]
```

Locomotive is available.

#### 9.44.4.4 \_direction

```
xpressnet::DoubleHeaderInformation::_direction [private]
```

Locomotive direction.

#### 9.44.4.5 `_function0`

`xpressnet::DoubleHeaderInformation::_function0` [private]

Function 0.

#### 9.44.4.6 `_function1`

`xpressnet::DoubleHeaderInformation::_function1` [private]

Function 1.

#### 9.44.4.7 `_function10`

`xpressnet::DoubleHeaderInformation::_function10` [private]

Function 10.

#### 9.44.4.8 `_function11`

`xpressnet::DoubleHeaderInformation::_function11` [private]

Function 11.

#### 9.44.4.9 `_function12`

`xpressnet::DoubleHeaderInformation::_function12` [private]

Function 12.

#### 9.44.4.10 `_function2`

`xpressnet::DoubleHeaderInformation::_function2` [private]

Function 2.

#### 9.44.4.11 \_function3

xpressnet::DoubleHeaderInformation::\_function3 [private]

Function 3.

#### 9.44.4.12 \_function4

xpressnet::DoubleHeaderInformation::\_function4 [private]

Function 4.

#### 9.44.4.13 \_function5

xpressnet::DoubleHeaderInformation::\_function5 [private]

Function 5.

#### 9.44.4.14 \_function6

xpressnet::DoubleHeaderInformation::\_function6 [private]

Function 6.

#### 9.44.4.15 \_function7

xpressnet::DoubleHeaderInformation::\_function7 [private]

Function 7.

#### 9.44.4.16 \_function8

xpressnet::DoubleHeaderInformation::\_function8 [private]

Function 8.

#### 9.44.4.17 `_function9`

`xpressnet::DoubleHeaderInformation::_function9` [private]

Function 9.

#### 9.44.4.18 `_speed`

`xpressnet::DoubleHeaderInformation::_speed` [private]

Locomotive speed.

#### 9.44.4.19 `_speedstep`

`xpressnet::DoubleHeaderInformation::_speedstep` [private]

Locomotive speed step mode.

### 9.45 `xpressnet::DoubleHeaderMuError` Class Reference

Double header or MU error.

#### Public Member Functions

- [DoubleHeaderMuError](#) (name, e)  
*Constructor.*
- [Error](#) ()  
*Return error type code.*

#### Private Attributes

- [\\_error](#)  
*Error type.*

#### 9.45.1 Detailed Description

Double header or MU error.

#### Author

Robert Heller <heller@deepsoft.com>



## 9.45.2 Constructor & Destructor Documentation

### 9.45.2.1 DoubleHeaderMuError()

```
xpressnet::DoubleHeaderMuError::DoubleHeaderMuError (
    name ,
    e )
```

Constructor.

Parameters

<i>e</i>	Error type.
----------	-------------

## 9.45.3 Member Function Documentation

### 9.45.3.1 Error()

```
xpressnet::DoubleHeaderMuError::Error ( )
```

Return error type code.

## 9.45.4 Member Data Documentation

### 9.45.4.1 \_error

```
xpressnet::DoubleHeaderMuError::_error [private]
```

Error type.

## 9.46 CTCPanel::DoubleSlip Class Reference

Double Slip (turnout) object type.

## Public Member Functions

- [DoubleSlip](#) (name, \_ctcpanel, \_canvas,...)  
*Construct a [DoubleSlip](#) object.*
- [~DoubleSlip](#) ()  
*Clean up all data objects and free up all resources.*
- [getv](#) ()  
*Method to get our value (state).*
- [setv](#) (value)  
*Method to set out value (state).*
- [geti](#) (ind)  
*Method to get the state of one of our indicators (none).*
- [seti](#) (ind, value)  
*Method to set an indicator's state (none).*
- [invoke](#) ()  
*Method to invoke the switch.*

## Private Member Functions

- [\\_configureLabel](#) (option, value)  
*Method to update the label option.*

## Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*
- [state](#)  
*The state of the points.*

### 9.46.1 Detailed Description

Double Slip (turnout) object type.

These are on the schematic and represent a switch on the Schematic.

#### Parameters

<a href="#">_ctcpanel</a>	The <a href="#">CTCPanel</a> megawidget.
<a href="#">_canvas</a>	The schematic canvas to draw the switch on.

## Parameters

...	<p>Options:</p> <ul style="list-style-type: none"> <li>• -x The x coordinate of the object (readonly, default 0).</li> <li>• -y The y coordinate of the object (readonly, default 0).</li> <li>• -controlpoint The name of the control point this label is part of (readonly, default CP1).</li> <li>• -label The label of the switch (default "1").</li> <li>• -orientation The orientation (8-way) of the switch (readonly, default 0).</li> <li>• -flipped Whether or not the switch is flipped (readonly, default no).</li> <li>• -statecommand A command to run to get the switch's state (default {}).</li> <li>• -occupiedcommand A command to run to find out if the switch is occupied (default {}).</li> </ul>
-----	--

Defined coords terminals:

- MainL Mainline left.
- MainR Mainline right.
- AltL Alternitive line left.
- AltR Alternitive line right.

Defined values (states):

- Normal Points are aligned for the mainline.
- Reverse Points are aligned for the branchline.
- Unknown Point are not aligned for any route (eg the points are in motion).

Defined indicators: none.

## Author

Robert Heller <heller@deepsoft.com>

## 9.46.2 Constructor &amp; Destructor Documentation

## 9.46.2.1 DoubleSlip()

```
CTCPanel::DoubleSlip::DoubleSlip (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a [DoubleSlip](#) object.

## Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the <a href="#">DoubleSlip</a> on.
<code>...</code>	Option list.

**9.46.2.2** `~DoubleSlip()`

```
CTCPanel::DoubleSlip::~~DoubleSlip ( )
```

Clean up all data objects and free up all resources.

**9.46.3** Member Function Documentation**9.46.3.1** `_configureLabel()`

```
CTCPanel::DoubleSlip::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

**9.46.3.2** `geti()`

```
CTCPanel::DoubleSlip::geti (
    ind )
```

Method to get the state of one of our indicators (none).

**9.46.3.3** `getv()`

```
CTCPanel::DoubleSlip::getv ( )
```

Method to get our value (state).

#### 9.46.3.4 invoke()

```
CTCPanel::DoubleSlip::invoke ( )
```

Method to invoke the switch.

#### 9.46.3.5 seti()

```
CTCPanel::DoubleSlip::seti (
    ind ,
    value )
```

Method to set an indicator's state (none).

#### 9.46.3.6 setv()

```
CTCPanel::DoubleSlip::setv (
    value )
```

Method to set out value (state).

##### Parameters

<i>value</i>	The new state to set.
--------------	-----------------------

### 9.46.4 Member Data Documentation

#### 9.46.4.1 canvas

```
CTCPanel::DoubleSlip::canvas [private]
```

The canvas component (parent widget component).

#### 9.46.4.2 ctcpnl

```
CTCPanel::DoubleSlip::ctcpnl [private]
```

The CTC Panel component (parent widget).

#### 9.46.4.3 state

```
CTCPanel::DoubleSlip::state [private]
```

The state of the points.

### 9.47 Parsers::TrackGraph::EdgeValues Struct Reference

Uncompressed graph edge values.

#### Public Member Functions

- [EdgeValues](#) (int \_index=-1, float \_x=0.0, float \_y=0.0, float \_a=0.0, float \_length=0.0)  
*Default constructor.*

#### Public Attributes

- int [index](#)  
*Index of next segment.*
- float [x](#)  
*X value of edge.*
- float [y](#)  
*Y value of edge.*
- float [a](#)  
*A value of edge.*
- float [length](#)  
*Track length from opposite edge.*

#### 9.47.1 Detailed Description

Uncompressed graph edge values.

#### Author

Robert Heller <heller@deepsoft.com>

## 9.47.2 Constructor & Destructor Documentation

### 9.47.2.1 EdgeValues()

```
Parsers::TrackGraph::EdgeValues::EdgeValues (
    int _index = -1,
    float _x = 0.0,
    float _y = 0.0,
    float _a = 0.0,
    float _length = 0.0 ) [inline]
```

Default constructor.

## 9.47.3 Member Data Documentation

### 9.47.3.1 a

```
float Parsers::TrackGraph::EdgeValues::a
```

A value of edge.

### 9.47.3.2 index

```
int Parsers::TrackGraph::EdgeValues::index
```

Index of next segment.

### 9.47.3.3 length

```
float Parsers::TrackGraph::EdgeValues::length
```

Track length from opposite edge.

## 9.47.3.4 x

```
float Parsers::TrackGraph::EdgeValues::x
```

X value of edge.

## 9.47.3.5 y

```
float Parsers::TrackGraph::EdgeValues::y
```

Y value of edge.

## 9.48 CTCPanel::EndBumper Class Reference

End Bumper object type.

### Public Member Functions

- [EndBumper](#) (name, \_ctcpanel, \_canvas,...)  
*Construct a [ScissorCrossover](#) object.*
- [~EndBumper](#) ()  
*Clean up all data objects and free up all resources.*
- [setv](#) (value)  
*Method to set out value (state).*
- [geti](#) (ind)  
*Method to get the state of one of our indicators (none).*
- [seti](#) (ind, value)  
*Method to set an indicator's state (none).*
- [invoke](#) ()  
*Method to invoke the [EndBumper](#).*

### Private Member Functions

- [\\_configureLabel](#) (option, value)  
*Method to update the label option.*

### Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*

### 9.48.1 Detailed Description

End Bumper object type.

These are on the schematic and represent an End Bumper on the Schematic.



## Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the switch on.
<code>...</code>	Options: <ul style="list-style-type: none"> <li>• <code>-x</code> The x coordinate of the object (readonly, default 0).</li> <li>• <code>-y</code> The y coordinate of the object (readonly, default 0).</li> <li>• <code>-controlpoint</code> The name of the control point this label is part of (readonly, default CP1).</li> <li>• <code>-label</code> The label of the switch (default "1").</li> <li>• <code>-position</code> The position of the label (readonly, default below).</li> <li>• <code>-orientation</code> The orientation (8-way) of the switch (readonly, default 0).</li> <li>• <code>-flipped</code> Whether or not the switch is flipped (readonly, default no).</li> <li>• <code>-occupiedcommand</code> A command to run to find out if the switch is occupied (default {}).</li> </ul>

Defined coords terminals:

- E End Point

Defined indicators: none.

## Author

Robert Heller <heller@deepsoft.com>

## 9.48.2 Constructor &amp; Destructor Documentation

## 9.48.2.1 EndBumper()

```
CTCPanel::EndBumper::EndBumper (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a [ScissorCrossover](#) object.

## Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the <a href="#">ScissorCrossover</a> on.
<code>...</code>	Option list.

**9.48.2.2** `~EndBumper()`

```
CTCPanel::EndBumper::~~EndBumper ( )
```

Clean up all data objects and free up all resources.

**9.48.3** Member Function Documentation**9.48.3.1** `_configureLabel()`

```
CTCPanel::EndBumper::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

**9.48.3.2** `geti()`

```
CTCPanel::EndBumper::geti (
    ind )
```

Method to get the state of one of our indicators (none).

**9.48.3.3** `invoke()`

```
CTCPanel::EndBumper::invoke ( )
```

Method to invoke the [EndBumper](#).

#### 9.48.3.4 seti()

```
CTCPanel::EndBumper::seti (
    ind ,
    value )
```

Method to set an indicator's state (none).

#### 9.48.3.5 setv()

```
CTCPanel::EndBumper::setv (
    value )
```

Method to set out value (state).

##### Parameters

<i>value</i>	The new state to set.
--------------	-----------------------

### 9.48.4 Member Data Documentation

#### 9.48.4.1 canvas

```
CTCPanel::EndBumper::canvas [private]
```

The canvas component (parent widget component).

#### 9.48.4.2 ctcpnl

```
CTCPanel::EndBumper::ctcpnl [private]
```

The CTC Panel component (parent widget).

## 9.49 TTSupport::eqstr Struct Reference

```
#include <TimeTableSystem.h>
```

## Public Member Functions

- bool [operator\(\)](#) (const char \*s1, const char \*s2) const

### 9.49.1 Member Function Documentation

#### 9.49.1.1 [operator\(\)](#)()

```
bool TTSupport::eqstr::operator() (
    const char * s1,
    const char * s2 ) const [inline]
```

## 9.50 Icc::EventID Class Reference

An event id structure.

## Public Member Functions

- [EventID](#) (name,...)  
*Constructor: create the event id.*

## Static Public Member Functions

- static [validate](#) (object)  
*Validation method.*

## Private Member Functions

- [\\_setEventID](#) (option, value)  
*Set (as in configure) the event ID.*
- [\\_getEventID](#) (option)  
*Get (as in cget) the event id.*

## Private Attributes

- [\\_eventID](#)  
*Internal representation of an event id, as an 8 byte list.*

## Static Private Attributes

- static [EVENTIDFMT](#)  
*Event ID format string.*

### 9.50.1 Detailed Description

An event id structure.

## Parameters

...	<p>The options:</p> <ul style="list-style-type: none"> <li>• -eventidstring The event ID as a string.</li> <li>• -eventidlist The event ID as a list.</li> </ul>
-----	--

## 9.50.2 Constructor &amp; Destructor Documentation

## 9.50.2.1 EventID()

```
lcc::EventID::EventID (
    name ,
    ... )
```

Constuctor: create the event id.

Create an eventid structure.

## Parameters

<i>name</i>	The name of the structure.
...	<p>The options:</p> <ul style="list-style-type: none"> <li>• -eventidstring The event ID as a string.</li> <li>• -eventidlist The event ID as a list.</li> </ul>

## 9.50.3 Member Function Documentation

## 9.50.3.1 \_getEventID()

```
lcc::EventID::_getEventID (
    option ) [private]
```

Get (as in cget) the event id.

The event id is returned, either as a string or a list.

**Parameters**

<i>option</i>	Either -eventidstring or -eventidlist. If it is -eventidstring, the event id list is formatted as a string. If it is -eventidlist the event id list is just returned.
---------------	---

**Returns**

The eventid as a string or a list.

**9.50.3.2 \_setEventID()**

```
lcc::EventID::_setEventID (
    option ,
    value ) [private]
```

Set (as in configure) the event ID.

Converts the value argument to the internal representation of the event id,

**Parameters**

<i>option</i>	Either -eventidstring or -eventidlist. If it is -eventidstring, the string is parsed and converted to a list of 8 bytes. If it is -eventidlist the list is just stored.
<i>value</i>	Either an eventidstring or an eventidlist.

**9.50.3.3 validate()**

```
static lcc::EventID::validate (
    object ) [static]
```

Validation method.

Validate [EventID](#) objects.

**Parameters**

<i>object</i>	The object to type check.
---------------	---------------------------

**Returns**

The object or raise an error.

## 9.50.4 Member Data Documentation

### 9.50.4.1 \_eventID

```
lcc::EventID::_eventID [private]
```

Internal representation of an event id, as an 8 byte list.

### 9.50.4.2 EVENTIDFMT

```
lcc::EventID::EVENTIDFMT [static], [private]
```

Event ID format string.

Event format string, used for both format and scan.

## 9.51 lcc::EventID\_or\_null Class Reference

An [EventID](#) or empty string.

### Static Public Member Functions

- static [validate](#) (value)  
*Validate a possible [EventID](#), but allow a null string value.*

### 9.51.1 Detailed Description

An [EventID](#) or empty string.

## 9.51.2 Member Function Documentation

### 9.51.2.1 validate()

```
static lcc::EventID_or_null::validate (  
    value ) [static]
```

Validate a possible [EventID](#), but allow a null string value.

## Parameters

<i>value</i>	The value to validate.
--------------	------------------------

## 9.52 Icc::EventLog Class Reference

Event received log, with event sender.

### Public Member Functions

- [EventLog](#) (name,...)  
*Construct an [EventLog](#) widget.*
- [open](#) ()  
*Open window.*
- [eventReceived](#) (eventid)  
*Log a received event.*

### Private Member Functions

- [\\_sendtheevent](#) ()  
*Send an event.*
- [\\_close](#) ()  
*Close the window.*
- [\\_clear](#) ()  
*Clear the log.*

### Private Attributes

- [logscroll](#)  
*Log Scroll Widget.*
- [logtext](#)  
*Log text Widget (readonly).*
- [sendevent](#)  
*Send event entry.*

### 9.52.1 Detailed Description

Event received log, with event sender.

Options:

- -transport The transport to use.



## 9.52.2 Constructor & Destructor Documentation

### 9.52.2.1 EventLog()

```
lcc::EventLog::EventLog (
    name ,
    ... )
```

Construct an [EventLog](#) widget.

This is a toplevel window with a scrolling log of received events. There is also an entry to send an event.

#### Parameters

...	Options: <ul style="list-style-type: none"><li>• -transport The transport to use.</li></ul>
-----	---

## 9.52.3 Member Function Documentation

### 9.52.3.1 \_clear()

```
lcc::EventLog::_clear ( ) [private]
```

Clear the log.

### 9.52.3.2 \_close()

```
lcc::EventLog::_close ( ) [private]
```

Close the window.

#### 9.52.3.3 `_sendtheevent()`

```
lcc::EventLog::_sendtheevent ( ) [private]
```

Send an event.

#### 9.52.3.4 `eventReceived()`

```
lcc::EventLog::eventReceived (
    eventid )
```

Log a received event.

##### Parameters

<i>eventid</i>	<a href="#">EventID</a> object to log.
----------------	--

#### 9.52.3.5 `open()`

```
lcc::EventLog::open ( )
```

Open window.

### 9.52.4 Member Data Documentation

#### 9.52.4.1 `logscroll`

```
lcc::EventLog::logscroll [private]
```

Log Scroll Widget.

#### 9.52.4.2 `logtext`

```
lcc::EventLog::logtext [private]
```

Log text Widget (readonly).

#### 9.52.4.3 sendevent

```
lcc::EventLog::sendevent [private]
```

Send event entry.

## 9.53 lcc::EventReceived Class Reference

Display a received event.

### Public Member Functions

- [EventReceived](#) (name,...)  
*Construct an [EventReceived](#) dialog.*
- [\\_Close](#) ()  
*Close and destroy the dialog box.*

### Private Attributes

- [eventid](#)  
*LabelEntry (RO) containing the eventId.*

### 9.53.1 Detailed Description

Display a received event.

Options:

- -eventid The event id – this is required.

### 9.53.2 Constructor & Destructor Documentation

#### 9.53.2.1 EventReceived()

```
lcc::EventReceived::EventReceived (  
    name ,  
    ... )
```

Construct an [EventReceived](#) dialog.

**Parameters**

<i>name</i>	The widget path.
...	The options: <ul style="list-style-type: none"><li>• -eventid The event id – this is required.</li></ul>

**9.53.3 Member Function Documentation****9.53.3.1 \_Close()**

```
lcc::EventReceived::_Close ( )
```

Close and destroy the dialog box.

**9.53.4 Member Data Documentation****9.53.4.1 eventId**

```
lcc::EventReceived::eventId [private]
```

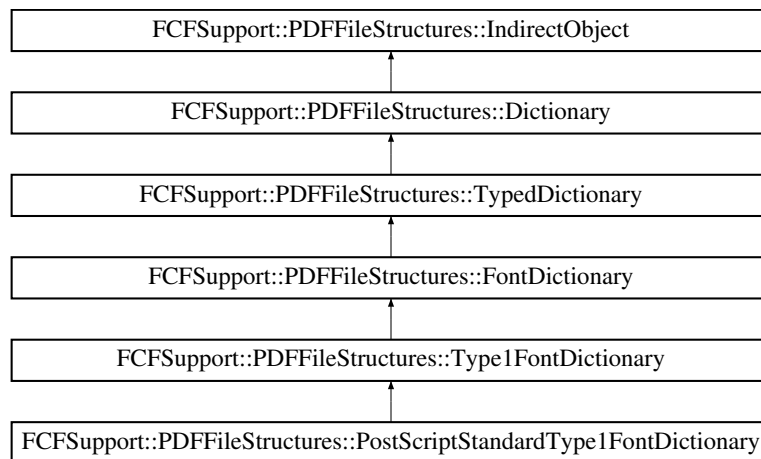
LabelEntry (RO) containing the eventId.

## 9.54 FCFSupport::PDFFileStructures::FontDictionary Class Reference

A Font dictionary object.

```
#include <PDFPrinterSupport.h>
```

Inheritance diagram for FCFSupport::PDFFileStructures::FontDictionary:



### Public Member Functions

- [FontDictionary](#) (const string subtype, unsigned long int objNum=0L, unsigned short int genNum=0, CrossReferenceTable \*tab=NULL)  
*Constructor.*
- [~FontDictionary](#) ()  
*Destructor.*

### Protected Member Functions

- ostream & [WriteFontType](#) (ostream &stream) const  
*Write the font's subtype.*
- virtual ostream & [WriteDictionaryElements](#) (ostream &stream) const  
*Write this dictionary's elements.*

### Private Attributes

- string [subType](#)  
*The type of the font.*

### 9.54.1 Detailed Description

A Font dictionary object.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.54.2 Constructor & Destructor Documentation

#### 9.54.2.1 FontDictionary()

```
FCFSupport::PDFFileStructures::FontDictionary::FontDictionary (
    const string subtype,
    unsigned long int objNum = 0L,
    unsigned short int genNum = 0,
    CrossReferenceTable * tab = NULL ) [inline]
```

Constructor.

Create a generic font dictionary.

#### Parameters

<i>subtype</i>	The type of the font.
<i>objNum</i>	The next object number.
<i>genNum</i>	The generation number.
<i>tab</i>	The cross reference table we are in.

#### 9.54.2.2 ~FontDictionary()

```
FCFSupport::PDFFileStructures::FontDictionary::~FontDictionary ( ) [inline]
```

Destructor.

### 9.54.3 Member Function Documentation

#### 9.54.3.1 WriteDictionaryElements()

```
virtual ostream& FCFSupport::PDFFileStructures::FontDictionary::WriteDictionaryElements (
    ostream & stream ) const [inline], [protected], [virtual]
```

Write this dictionary's elements.

Start with its type.

##### Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

Reimplemented from [FCFSupport::PDFFileStructures::TypedDictionary](#).

Reimplemented in [FCFSupport::PDFFileStructures::Type1FontDictionary](#).

References lcc::stream.

#### 9.54.3.2 WriteFontType()

```
ostream& FCFSupport::PDFFileStructures::FontDictionary::WriteFontType (
    ostream & stream ) const [inline], [protected]
```

Write the font's subtype.

##### Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

### 9.54.4 Member Data Documentation

#### 9.54.4.1 subType

```
string FCFSupport::PDFFileStructures::FontDictionary::subType [private]
```

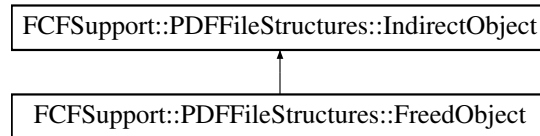
The type of the font.

## 9.55 FCFSupport::PDFFileStructures::FreedObject Class Reference

A deleted indirect object.

```
#include <PDFPrinterSupport.h>
```

Inheritance diagram for FCFSupport::PDFFileStructures::FreedObject:



### Public Member Functions

- [FreedObject](#) (unsigned long int objNum, unsigned short int genNum, [CrossReferenceTable](#) \*tab)  
*Constructor.*
- virtual [~FreedObject](#) ()  
*Destructor.*
- virtual ostream & [WriteDirect](#) (ostream &stream) const  
*Dummy function for direct writing (should never be called).*

### 9.55.1 Detailed Description

A deleted indirect object.

Just a place holder in the linked list of freed indirect objects.

Author

Robert Heller <heller@deepsoft.com>

### 9.55.2 Constructor & Destructor Documentation

#### 9.55.2.1 FreedObject()

```
FCFSupport::PDFFileStructures::FreedObject::FreedObject (
    unsigned long int objNum,
    unsigned short int genNum,
    CrossReferenceTable * tab ) [inline]
```

Constructor.

Create a freed object.



## Parameters

<i>objNum</i>	The next object number.
<i>genNum</i>	The generation number.
<i>tab</i>	The cross reference table we are in.

## 9.55.2.2 ~FreedObject()

```
virtual FCFSupport::PDFFileStructures::FreedObject::~~FreedObject ( ) [inline], [virtual]
```

Destructor.

Clean everything up.

## 9.55.3 Member Function Documentation

## 9.55.3.1 WriteDirect()

```
virtual ostream& FCFSupport::PDFFileStructures::FreedObject::WriteDirect (
    ostream & stream ) const [inline], [virtual]
```

Dummy function for direct writing (should never be called).

## Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

Implements [FCFSupport::PDFFileStructures::IndirectObject](#).

References [lcc::stream](#).

## 9.56 xpressnet::FunctionStatus Class Reference

Function status.

## Public Member Functions

- [FunctionStatus](#) (name, s0, s1, s2, s3, s4, s5, s6, s7, s8, s9, s10, s11, s12)  
*Constructor.*
- [Status](#) (f)  
*Return selected status flag.*

## Private Attributes

- [\\_status0](#)  
*Status 0.*
- [\\_status1](#)  
*Status 1.*
- [\\_status2](#)  
*Status 2.*
- [\\_status3](#)  
*Status 3.*
- [\\_status4](#)  
*Status 4.*
- [\\_status5](#)  
*Status 5.*
- [\\_status6](#)  
*Status 6.*
- [\\_status7](#)  
*Status 7.*
- [\\_status8](#)  
*Status 8.*
- [\\_status9](#)  
*Status 9.*
- [\\_status10](#)  
*Status 10.*
- [\\_status11](#)  
*Status 11.*
- [\\_status12](#)  
*Status 12.*

### 9.56.1 Detailed Description

Function status.

#### Author

Robert Heller <heller@deepsoft.com>

## 9.56.2 Constructor & Destructor Documentation

### 9.56.2.1 FunctionStatus()

```
xpressnet::FunctionStatus::FunctionStatus (
    name ,
    s0 ,
    s1 ,
    s2 ,
    s3 ,
    s4 ,
    s5 ,
    s6 ,
    s7 ,
    s8 ,
    s9 ,
    s10 ,
    s11 ,
    s12 )
```

Constructor.

#### Parameters

<i>s0</i>	Function 0 is monemtary flag.
<i>s1</i>	Function 1 is monemtary flag.
<i>s2</i>	Function 2 is monemtary flag.
<i>s3</i>	Function 3 is monemtary flag.
<i>s4</i>	Function 4 is monemtary flag.
<i>s5</i>	Function 5 is monemtary flag.
<i>s6</i>	Function 6 is monemtary flag.
<i>s7</i>	Function 7 is monemtary flag.
<i>s8</i>	Function 8 is monemtary flag.
<i>s9</i>	Function 9 is monemtary flag.
<i>s10</i>	Function 10 is monemtary flag.
<i>s11</i>	Function 11 is monemtary flag.
<i>s12</i>	Function 12 is monemtary flag.

## 9.56.3 Member Function Documentation

#### 9.56.3.1 Status()

```
xpressnet::FunctionStatus::Status (
    f )
```

Return selected status flag.

##### Parameters

<i>f</i>	Function whose status to return.
----------	----------------------------------

### 9.56.4 Member Data Documentation

#### 9.56.4.1 \_status0

```
xpressnet::FunctionStatus::_status0 [private]
```

Status 0.

#### 9.56.4.2 \_status1

```
xpressnet::FunctionStatus::_status1 [private]
```

Status 1.

#### 9.56.4.3 \_status10

```
xpressnet::FunctionStatus::_status10 [private]
```

Status 10.

#### 9.56.4.4 \_status11

```
xpressnet::FunctionStatus::_status11 [private]
```

Status 11.

#### 9.56.4.5 \_status12

xpressnet::FunctionStatus::\_status12 [private]

Status 12.

#### 9.56.4.6 \_status2

xpressnet::FunctionStatus::\_status2 [private]

Status 2.

#### 9.56.4.7 \_status3

xpressnet::FunctionStatus::\_status3 [private]

Status 3.

#### 9.56.4.8 \_status4

xpressnet::FunctionStatus::\_status4 [private]

Status 4.

#### 9.56.4.9 \_status5

xpressnet::FunctionStatus::\_status5 [private]

Status 5.

#### 9.56.4.10 \_status6

xpressnet::FunctionStatus::\_status6 [private]

Status 6.

#### 9.56.4.11 `_status7`

`xpressnet::FunctionStatus::_status7` [private]

Status 7.

#### 9.56.4.12 `_status8`

`xpressnet::FunctionStatus::_status8` [private]

Status 8.

#### 9.56.4.13 `_status9`

`xpressnet::FunctionStatus::_status9` [private]

Status 9.

## 9.57 `linuxgpio::GpioInputActiveHigh` Class Reference

Input pin, active high (high is true).

### Public Member Functions

- [GpioInputActiveHigh](#) (name,...)  
*Constructor, used to set up the GPIO pin.*
- [~GpioInputActiveHigh](#) ()  
*Destructor.*

### Private Attributes

- [basepin](#)  
*The base pin.*

### 9.57.1 Detailed Description

Input pin, active high (high is true).

## Parameters

<i>name</i>	Name of the pin.
...	Options: <ul style="list-style-type: none"><li>• -pinnumber The pin number, readonly, defaults to 0 and can be any positive integer.</li></ul>

## Author

Robert Heller <heller@deepsoft.com>

## 9.57.2 Constructor & Destructor Documentation

### 9.57.2.1 GpioInputActiveHigh()

```
linuxgpio::GpioInputActiveHigh::GpioInputActiveHigh (
    name ,
    ... )
```

Constructor, used to set up the GPIO pin.

The pin number is written to the export control file and then the pin's direction control file is computed and the pin's direction is written.

## Parameters

<i>name</i>	The name of the pin.
...	Options: <ul style="list-style-type: none"><li>• -pinnumber The pin number, readonly, defaults to 0 and can be any positive integer.</li></ul>

## Author

Robert Heller <heller@deepsoft.com>

### 9.57.2.2 ~GpioInputActiveHigh()

```
linuxgpio::GpioInputActiveHigh::~~GpioInputActiveHigh ( )
```

Destructor.

Unexport the pin.

## 9.57.3 Member Data Documentation

### 9.57.3.1 basepin

```
linuxgpio::GpioInputActiveHigh::basepin [private]
```

The base pin.

## 9.58 linuxgpio::GpioInputActiveLow Class Reference

Input pin, active low (low is true).

### Public Member Functions

- [GpioInputActiveLow](#) (name,...)  
*Constructor, used to set up the GPIO pin.*
- [~GpioInputActiveLow](#) ()  
*Destructor.*
- [Get](#) ()  
*Get the pin's logic state.*

### Private Attributes

- [basepin](#)  
*The base pin.*

### 9.58.1 Detailed Description

Input pin, active low (low is true).



## Parameters

<i>name</i>	Name of the pin.
...	Options: <ul style="list-style-type: none"><li>• -pinnumber The pin number, readonly, defaults to 0 and can be any positive integer.</li></ul>

## Author

Robert Heller <heller@deepsoft.com>

## 9.58.2 Constructor & Destructor Documentation

### 9.58.2.1 GpioInputActiveLow()

```
linuxgpio::GpioInputActiveLow::GpioInputActiveLow (
    name ,
    ... )
```

Constructor, used to set up the GPIO pin.

The pin number is written to the export control file and then the pin's direction control file is computed and the pin's direction is written.

## Parameters

<i>name</i>	The name of the pin.
...	Options: <ul style="list-style-type: none"><li>• -pinnumber The pin number, readonly, defaults to 0 and can be any positive integer.</li></ul>

## Author

Robert Heller <heller@deepsoft.com>

### 9.58.2.2 ~GpioInputActiveLow()

```
linuxgpio::GpioInputActiveLow::~~GpioInputActiveLow ( )
```

Destructor.

Unexport the pin.

## 9.58.3 Member Function Documentation

### 9.58.3.1 Get()

```
linuxgpio::GpioInputActiveLow::Get ( )
```

Get the pin's logic state.

Returns

The pin state (low is true, high is false).

## 9.58.4 Member Data Documentation

### 9.58.4.1 basepin

```
linuxgpio::GpioInputActiveLow::basepin [private]
```

The base pin.

## 9.59 linuxgpio::GpioOutputSafeHigh Class Reference

Output pin, initialized to high.

### Public Member Functions

- [GpioOutputSafeHigh](#) (name,...)  
*Constructor, used to set up the GPIO pin.*
- [~GpioOutputSafeHigh](#) ()  
*Destructor.*

### Private Attributes

- [basepin](#)  
*The base pin.*

### 9.59.1 Detailed Description

Output pin, initialized to high.

## Parameters

<i>name</i>	Name of the pin.
...	Options: <ul style="list-style-type: none"><li>• -pinnumber The pin number, readonly, defaults to 0 and can be any positive integer.</li></ul>

## Author

Robert Heller <heller@deepsoft.com>

## 9.59.2 Constructor & Destructor Documentation

### 9.59.2.1 GpioOutputSafeHigh()

```
linuxgpio::GpioOutputSafeHigh::GpioOutputSafeHigh (
    name ,
    ... )
```

Constructor, used to set up the GPIO pin.

The pin number is written to the export control file and then the pin's direction control file is computed and the pin's direction is written.

## Parameters

<i>name</i>	The name of the pin.
...	Options: <ul style="list-style-type: none"><li>• -pinnumber The pin number, readonly, defaults to 0 and can be any positive integer.</li></ul>

## Author

Robert Heller <heller@deepsoft.com>

### 9.59.2.2 ~GpioOutputSafeHigh()

```
linuxgpio::GpioOutputSafeHigh::~~GpioOutputSafeHigh ( )
```

Destructor.

Unexport the pin.

## 9.59.3 Member Data Documentation

### 9.59.3.1 basepin

```
linuxgpio::GpioOutputSafeHigh::basepin [private]
```

The base pin.

## 9.60 linuxgpio::GpioOutputSafeHighInvert Class Reference

Output pin, initialized to high, inverted.

### Public Member Functions

- [GpioOutputSafeHighInvert](#) (name,...)  
*Constructor, used to set up the GPIO pin.*
- [~GpioOutputSafeHighInvert](#) ()  
*Destructor.*
- [Set](#) ()  
*Set the pin to true (logic low).*
- [Clr](#) ()  
*Set the pin to false (logic high).*
- [Get](#) ()  
*Get the pin's logic state.*

### Private Attributes

- [basepin](#)  
*The base pin.*

### 9.60.1 Detailed Description

Output pin, initialized to high, inverted.

## Parameters

<i>name</i>	Name of the pin.
...	Options: <ul style="list-style-type: none"><li>• -pinnumber The pin number, readonly, defaults to 0 and can be any positive integer.</li></ul>

## Author

Robert Heller <heller@deepsoft.com>

## 9.60.2 Constructor & Destructor Documentation

### 9.60.2.1 GpioOutputSafeHighInvert()

```
linuxgpio::GpioOutputSafeHighInvert::GpioOutputSafeHighInvert (
    name ,
    ... )
```

Constructor, used to set up the GPIO pin.

The pin number is written to the export control file and then the pin's direction control file is computed and the pin's direction is written.

## Parameters

<i>name</i>	The name of the pin.
...	Options: <ul style="list-style-type: none"><li>• -pinnumber The pin number, readonly, defaults to 0 and can be any positive integer.</li></ul>

## Author

Robert Heller <heller@deepsoft.com>

### 9.60.2.2 ~GpioOutputSafeHighInvert()

```
linuxgpio::GpioOutputSafeHighInvert::~~GpioOutputSafeHighInvert ( )
```

Destructor.

Unexport the pin.

## 9.60.3 Member Function Documentation

### 9.60.3.1 Clr()

```
linuxgpio::GpioOutputSafeHighInvert::Clr ( )
```

Set the pin to false (logic high).

### 9.60.3.2 Get()

```
linuxgpio::GpioOutputSafeHighInvert::Get ( )
```

Get the pin's logic state.

#### Returns

The pin state (low is true, high is false).

### 9.60.3.3 Set()

```
linuxgpio::GpioOutputSafeHighInvert::Set ( )
```

Set the pin to true (logic low).

## 9.60.4 Member Data Documentation

## 9.60.4.1 basepin

```
linuxgpio::GpioOutputSafeHighInvert::basepin [private]
```

The base pin.

## 9.61 linuxgpio::GpioOutputSafeLow Class Reference

Output pin, initialized to low.

### Public Member Functions

- [GpioOutputSafeLow](#) (name,...)  
*Constructor, used to set up the GPIO pin.*
- [~GpioOutputSafeLow](#) ()  
*Destructor.*

### Private Attributes

- [basepin](#)  
*The base pin.*

### 9.61.1 Detailed Description

Output pin, initialized to low.

#### Parameters

<i>name</i>	Name of the pin.
...	Options: <ul style="list-style-type: none"> <li>• -pinnumber The pin number, readonly, defaults to 0 and can be any positive integer.</li> </ul>

#### Author

Robert Heller <heller@deepsoft.com>

### 9.61.2 Constructor & Destructor Documentation

### 9.61.2.1 GpioOutputSafeLow()

```
linuxgpio::GpioOutputSafeLow::GpioOutputSafeLow (
    name ,
    ... )
```

Constructor, used to set up the GPIO pin.

The pin number is written to the export control file and then the pin's direction control file is computed and the pin's direction is written.

#### Parameters

<i>name</i>	The name of the pin.
...	Options: <ul style="list-style-type: none"> <li>-pinnumber The pin number, readonly, defaults to 0 and can be any positive integer.</li> </ul>

#### Author

Robert Heller <heller@deepsoft.com>

```
puts stderr "$type create $self $args"
```

### 9.61.2.2 ~GpioOutputSafeLow()

```
linuxgpio::GpioOutputSafeLow::~~GpioOutputSafeLow ( )
```

Destructor.

Unexport the pin.

## 9.61.3 Member Data Documentation

### 9.61.3.1 basepin

```
linuxgpio::GpioOutputSafeLow::basepin [private]
```

The base pin.



## 9.62 linuxgpio::GpioOutputSafeLowInverted Class Reference

Output pin, initialized to low, with inverted logic.

### Public Member Functions

- [GpioOutputSafeLowInverted](#) (name,...)  
*Constructor, used to set up the GPIO pin.*
- [Set](#) ()  
*Set the pin to true (logic low).*
- [Clr](#) ()  
*Set the pin to false (logic high).*
- [Get](#) ()  
*Get the pin's logic state.*
- [~GpioOutputSafeLowInverted](#) ()  
*Destructor.*

### Private Attributes

- [basepin](#)  
*The base pin.*

### 9.62.1 Detailed Description

Output pin, initialized to low, with inverted logic.

#### Parameters

<i>name</i>	Name of the pin.
...	Options: <ul style="list-style-type: none"><li>• -pinnumber The pin number, readonly, defaults to 0 and can be any positive integer.</li></ul>

#### Author

Robert Heller <heller@deepsoft.com>

### 9.62.2 Constructor & Destructor Documentation

### 9.62.2.1 GpioOutputSafeLowInverted()

```
linuxgpio::GpioOutputSafeLowInverted::GpioOutputSafeLowInverted (
    name ,
    ... )
```

Constructor, used to set up the GPIO pin.

The pin number is written to the export control file and then the pin's direction control file is computed and the pin's direction is written.

#### Parameters

<i>name</i>	The name of the pin.
...	Options: <ul style="list-style-type: none"><li>• -pinnumber The pin number, readonly, defaults to 0 and can be any positive integer.</li></ul>

#### Author

Robert Heller <heller@deepsoft.com>

### 9.62.2.2 ~GpioOutputSafeLowInverted()

```
linuxgpio::GpioOutputSafeLowInverted::~~GpioOutputSafeLowInverted ( )
```

Destructor.

Unexport the pin.

## 9.62.3 Member Function Documentation

### 9.62.3.1 Clr()

```
linuxgpio::GpioOutputSafeLowInverted::Clr ( )
```

Set the pin to false (logic high).

### 9.62.3.2 Get()

```
linuxgpio::GpioOutputSafeLowInverted::Get ( )
```

Get the pin's logic state.

#### Returns

The pin state (low is true, high is false).

### 9.62.3.3 Set()

```
linuxgpio::GpioOutputSafeLowInverted::Set ( )
```

Set the pin to true (logic low).

## 9.62.4 Member Data Documentation

### 9.62.4.1 basepin

```
linuxgpio::GpioOutputSafeLowInverted::basepin [private]
```

The base pin.

## 9.63 lcc::GridConnectMessage Class Reference

A Grid Connect formatted CAN message.

### Public Member Functions

- [GridConnectMessage](#) (name,...)  
*Constructor: create a Grid Connect Message object.*
- [setHeader](#) (header)  
*Set the header.*
- [setByte](#) (val, n)  
*Set a data byte.*
- [setHexDigit](#) (val, n)  
*Set a hex digit.*

## Private Member Functions

- [\\_set\\_extended](#) (opt, extended)  
*Configure method for the -extended option.*
- [\\_get\\_extended](#) (opt)  
*CGet method for the -extended option.*
- [\\_set\\_rtr](#) (opt, rtr)  
*Configure method for the -rtr option.*
- [\\_get\\_rtr](#) (opt)  
*CGet method for the -rtr option.*
- [\\_copyCM](#) (option, m)  
*Configure method for the -canmessage option.*

### 9.63.1 Detailed Description

A Grid Connect formatted CAN message.

This is an ASCII formatted version of a CAN message, used by some USB connected CAN interface devices.

This class is used to convert from **binary** CAN Messages to **ASCII** Grid Connect messages. See [GridConnectReply](#) for converting from **ASCII** Grid Connect messages to **binary** CAN Messages.

Options:

- -canmessage A binary CANMessage to be converted to a Grid Connect message. A write only option.
- -extended A boolean flag to indicate if this is an extended protocol message. Default no.
- -rtr A boolean flag to indicate if this is a reply expected message. Default no.

Additional methods defined using the macros AbstractMessage and AbstractMRMessage include:

- getElement {n} – Get the nth data element.
- getNumDataElements {} – Get the number of data elements.
- setElement {n v} – Set the nth data element.
- setOpCode {i} – Set the opcode (byte 0).
- getOpCode {} – Get the opcode (byte 0).
- getOpCodeHex {} – Get the opcode (byte 0) in hex.
- setNeededMode {pMode} – Set the needed mode.
- getNeededMode {} – Get the needed mode.
- replyExpected {} – Returns reply expected flag.
- isBinary {} – Returns binary flag.
- setBinary {b} – Set the binary flag.
- setTimeout {t} – Set the timeout.

- `getTimeout {}` – Get the timeout.
- `setRetries {i}` – Set the number of retries.
- `getRetries {}` – Get the number of retries.
- `addIntAsThree {val offset}` – Insert an integer as three decimal digits (with leading 0s).
- `addIntAsTwoHex {val offset}` – Insert an integer as two hexadecimal digits (with leading 0s).
- `addIntAsThreeHex {val offset}` – Insert an integer as three hexadecimal digits (with leading 0s).
- `addIntAsFourHex {val offset}` – Insert an integer as four hexadecimal digits (with leading 0s).
- `setNumDataElements {n}` – Set the number of data bytes.
- `toString {}` – Return the data object as a string.

And these (private) instance variables:

- `_dataChars {}`
- `_nDataChars 0`
- `mNeededMode 0`
- `_isBinary false`
- `mTimeout 0`
- `mRetries 0`

And these (private) static variables:

- `SHORT_TIMEOUT 2000`
- `LONG_TIMEOUT 60000`

## 9.63.2 Constructor & Destructor Documentation

### 9.63.2.1 GridConnectMessage()

```
lcc::GridConnectMessage::GridConnectMessage (
    name ,
    ... )
```

Constructor: create a Grid Connect Message object.

Create a Grid Connect Message. Typically, a CANMessage is configured with the `-canmessage` option and then the `toString` method is used to get a printable Grid Connect Message string.

## Parameters

<i>name</i>	The name of the object.
...	<p>The options:</p> <ul style="list-style-type: none"><li>• -canmessage A binary CANMessage to be converted to a Grid Connect message. A write only option.</li><li>• -extended A boolean flag to indicate if this is an extended protocol message.</li><li>• -rtr A boolean flag to indicate if this is a reply expected message.</li></ul>

### 9.63.3 Member Function Documentation

#### 9.63.3.1 \_copyCM()

```
lcc::GridConnectMessage::_copyCM (
    option ,
    m ) [private]
```

Configure method for the -canmessage option.

Copies in a CANMessage and in the process formats a Grid Connect Message string.

## Parameters

<i>option</i>	Always -canmessage. Ignored.
<i>m</i>	A CANMessage object.

#### 9.63.3.2 \_get\_extended()

```
lcc::GridConnectMessage::_get_extended (
    opt ) [private]
```

CGet method for the -extended option.

Gets the extended flag character.

**Parameters**

<i>opt</i>	Always -extended. Ignored.
------------	----------------------------

**Returns**

A boolean flag indicating extendedness.

**9.63.3.3 \_get\_rtr()**

```
lcc::GridConnectMessage::_get_rtr (
    opt ) [private]
```

CGet method for the -rtr option.

Gets the rtr flag character.

**Parameters**

<i>opt</i>	Always -rtr. Ignored. boolean flag indicating rtrness.
------------	--

**9.63.3.4 \_set\_extended()**

```
lcc::GridConnectMessage::_set_extended (
    opt ,
    extended ) [private]
```

Configure method for the -extended option.

Sets the extended flag character.

**Parameters**

<i>opt</i>	Always -extended. Ignored.
<i>extended</i>	Boolean flag indicating extendedness.

**9.63.3.5 \_set\_rtr()**

```
lcc::GridConnectMessage::_set_rtr (
```

```
    opt ,  
    rtr ) [private]
```

Configure method for the -rtr option.

Sets the rtr flag character.

#### Parameters

<i>opt</i>	Always -rtr. Ignored.
<i>rtr</i>	Boolean flag indicating rtness.

#### 9.63.3.6 setByte()

```
lcc::GridConnectMessage::setByte (   
    val ,  
    n )
```

Set a data byte.

Stores a data byte as two hex digits.

#### Parameters

<i>val</i>	The data byte value, 0-255.
<i>n</i>	The data index, 0-7.

#### 9.63.3.7 setHeader()

```
lcc::GridConnectMessage::setHeader (   
    header )
```

Set the header.

Sets the header. The header is converted to hex digits and stored in the data buffer.

#### Parameters

<i>header</i>	The binary 29-bit header.
---------------	---------------------------



## 9.63.3.8 setHexDigit()

```
lcc::GridConnectMessage::setHexDigit (
    val ,
    n )
```

Set a hex digit.

Stores a single nibble (0-16) at the specified index as an ASCII hex digit.

## Parameters

<i>val</i>	The nibble (0-16) to store.
<i>n</i>	The data index.

## 9.64 lcc::GridConnectReply Class Reference

A Grid Connect formatted CAN message (reply).

## Public Member Functions

- [GridConnectReply](#) (name,...)  
*Constructor: create a [GridConnectReply](#) instance.*
- [createReply](#) ()  
*Convert to a **binary** [CanMessage](#) object.*
- [setElement](#) (n, v)  
*Set the element.*
- [maxSize](#) ()  
*Return the maximum size of a Grid Connect Message.*
- [setData](#) (d)  
*Set the data Copy the data bytes into the structure.*
- [getHeader](#) ()  
*Extract the header as a 29-bit integer.*
- [getNumBytes](#) ()  
*Return the number of data bytes.*
- [getBytes](#) (b)  
*Return a selected data byte.*
- [getHexDigit](#) (index)  
*Get one hexadecimal digit.*

## Private Member Functions

- [\\_get\\_extended](#) (opt)  
*CGet method for the -extended option.*
- [\\_get\\_rtr](#) (opt)  
*CGet method for the -rtr option.*
- [\\_copyGCM](#) (option, s)  
*Configure method for the -message option.*
- [basicFormatCheck](#) ()  
*Perform a basic format check.*

## Private Attributes

- [\\_RTRoffset](#)  
*The offset to the RTR flag.*

## Static Private Attributes

- static [MAXLEN](#)  
*The maximum length for a Grid Connect Message.*

### 9.64.1 Detailed Description

A Grid Connect formatted CAN message (reply).

This is an ASCII formatted version of a CAN message, used by some USB connected CAN interface devices.

This class is used to convert to **binary** CAN Messages from **ASCII** Grid Connect messages. See [GridConnectMessage](#) for converting to **ASCII** Grid Connect messages from **binary** CAN Messages.

Options:

- -extended A boolean flag to indicate if this is an extended protocol message. Readonly and not settable.
- -rtr A boolean flag to indicate if this is a reply expected message. Readonly and not settable.
- -message A received [GridConnectMessage](#) to be converted to a binary [CanMessage](#). Settable only.

Additional methods defined using the macros AbstractMessage and AbstractMRMessage include:

- getElement {n} – Get the nth data element.
- getNumDataElements {} – Get the number of data elements.
- setElement {n v} – Set the nth data element.
- setOpCode {i} – Set the opcode (byte 0).
- getOpCode {} – Get the opcode (byte 0).

- `getOpCodeHex {}` – Get the opcode (byte 0) in hex.
- `setNeededMode {pMode}` – Set the needed mode.
- `getNeededMode {}` – Get the needed mode.
- `replyExpected {}` – Returns reply expected flag.
- `isBinary {}` – Returns binary flag.
- `setBinary {b}` – Set the binary flag.
- `setTimeout {t}` – Set the timeout.
- `getTimeout {}` – Get the timeout.
- `setRetries {}` – Set the number of retries.
- `getRetries {}` – Get the number of retries.
- `addIntAsThree {val offset}` – Insert an integer as three decimal digits (with leading 0s).
- `addIntAsTwoHex {val offset}` – Insert an integer as two hexadecimal digits (with leading 0s).
- `addIntAsThreeHex {val offset}` – Insert an integer as three hexadecimal digits (with leading 0s).
- `addIntAsFourHex {val offset}` – Insert an integer as four hexadecimal digits (with leading 0s).
- `setNumDataElements {n}` – Set the number of data bytes.
- `toString {}` – Return the data object as a string.

And these (private) instance variables:

- `_dataChars {}`
- `_nDataChars 0`
- `mNeededMode 0`
- `_isBinary false`
- `mTimeout 0`
- `mRetries 0`

And these (private) static variables:

- `SHORT_TIMEOUT 2000`
- `LONG_TIMEOUT 60000`

### 9.64.2 Constructor & Destructor Documentation

### 9.64.2.1 GridConnectReply()

```
lcc::GridConnectReply::GridConnectReply (
    name ,
    ... )
```

Constructor: create a [GridConnectReply](#) instance.

A [GridConnectReply](#) object is created.

#### Parameters

<i>name</i>	The name of the new instance.
...	The options: <ul style="list-style-type: none"><li>• -message An optional Grid Connect Message string.</li></ul>

## 9.64.3 Member Function Documentation

### 9.64.3.1 \_copyGCM()

```
lcc::GridConnectReply::_copyGCM (
    option ,
    s ) [private]
```

Configure method for the -message option.

Send in an ASCII Grid Connect Message for conversion.

#### Parameters

<i>option</i>	Allways -message. Ignored.
<i>s</i>	The ASCII Grid Connect Message as a string.

### 9.64.3.2 \_get\_extended()

```
lcc::GridConnectReply::_get_extended (
    opt ) [private]
```

CGet method for the -extended option.

Gets the extended protocol flag for this message.

#### Parameters

<i>opt</i>	Allways -extended. Ignored.
------------	-----------------------------

#### Returns

The extended protocol flag for this message.

#### 9.64.3.3 \_get\_rtr()

```
lcc::GridConnectReply::_get_rtr (
    opt ) [private]
```

CGet method for the -rtr option.

Gets the reply flag for this message.

#### Parameters

<i>opt</i>	Allways -rtr. Ignored.
------------	------------------------

#### Returns

The reply flag for this message.

#### 9.64.3.4 basicFormatCheck()

```
lcc::GridConnectReply::basicFormatCheck ( ) [private]
```

Perform a basic format check.

Check for a basicly correct formatted string.

#### Returns

A boolean flag indicating that the message passed a basic format check.

#### 9.64.3.5 createReply()

```
lcc::GridConnectReply::createReply ( )
```

Convert to a **binary** [CanMessage](#) object.

Decode a Grid Connect Message into a binary [CanMessage](#) object.

##### Returns

A [CanMessage](#) object.

#### 9.64.3.6 getByte()

```
lcc::GridConnectReply::getByte (
    b )
```

Return a selected data byte.

##### Parameters

<i>b</i>	The index of the byte (0-7) to return.
----------	--

##### Returns

The data bytes or 0.

#### 9.64.3.7 getHeader()

```
lcc::GridConnectReply::getHeader ( )
```

Extract the header as a 29-bit integer.

Peel the hexadecimal digits between the simple/extended flag character and the reply/noreply character as a 29-bit CAN header word.

##### Returns

A 29-bit integer.

#### 9.64.3.8 getHexDigit()

```
lcc::GridConnectReply::getHexDigit (
    index )
```

Get one hexadecimal digit.

**Parameters**

<i>index</i>	The low-level data index of the nibble to return.
--------------	---

**Returns**

The nibble.

**9.64.3.9 getNumBytes()**

```
lcc::GridConnectReply::getNumBytes ( )
```

Return the number of data bytes.

**Returns**

The number of data bytes.

**9.64.3.10 maxSize()**

```
lcc::GridConnectReply::maxSize ( )
```

Return the maximum size of a Grid Connect Message.

**Returns**

The maximum size of a Grid Connect Message.

**9.64.3.11 setData()**

```
lcc::GridConnectReply::setData (
    d )
```

Set the data Copy the data bytes into the structure.

**Parameters**

<i>d</i>	A list of data bytes (characters).
----------	------------------------------------

### 9.64.3.12 setElement()

```
lcc::GridConnectReply::setElement (
    n ,
    v )
```

Set the element.

Set the element at the specified index.

#### Parameters

<i>n</i>	The index to set.
<i>v</i>	The value to set.

## 9.64.4 Member Data Documentation

### 9.64.4.1 \_RTRoffset

```
lcc::GridConnectReply::_RTRoffset [private]
```

The offset to the RTR flag.

### 9.64.4.2 MAXLEN

```
lcc::GridConnectReply::MAXLEN [static], [private]
```

The maximum length for a Grid Connect Message.

## 9.65 TTSupport::hash Struct Reference

Option hash map, used for Print options.

```
#include <TimeTableSystem.h>
```



## Public Member Functions

- `std::size_t operator()` (const char \*s) const

### 9.65.1 Detailed Description

Option hash map, used for Print options.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.65.2 Member Function Documentation

#### 9.65.2.1 operator()()

```
std::size_t TTSupport::hash::operator() (
    const char * s ) const [inline]
```

## 9.66 CTCPanel::HiddenBlock Class Reference

Hidden Block object type.

## Public Member Functions

- `HiddenBlock` (name, \_ctcpanel, \_canvas,...)  
*Construct a [HiddenBlock](#) object.*
- `~HiddenBlock` ()  
*Clean up all data objects and free up all resources.*
- `getv` ()  
*Method to get our value (state).*
- `setv` (value)  
*Method to set out value (state).*
- `geti` (ind)  
*Method to get the state of one of our indicators (none).*
- `seti` (ind, value)  
*Method to set an indicator's state (none).*
- `invoke` ()  
*Method to invoke the [HiddenBlock](#).*

## Private Member Functions

- [\\_configureLabel](#) (option, value)  
*Method to update the label option.*

## Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*

### 9.66.1 Detailed Description

Hidden Block object type.

These are on the schematic and represent a piece of track on the Schematic.

#### Parameters

<a href="#">_ctcpanel</a>	The <a href="#">CTCPanel</a> megawidget.
<a href="#">_canvas</a>	The schematic canvas to draw the <a href="#">HiddenBlock</a> on.
...	Options: <ul style="list-style-type: none"> <li>• -x1 The first x coordinate of the object (readonly, default 0).</li> <li>• -y1 The first y coordinate of the object (readonly, default 0).</li> <li>• -x2 The second x coordinate of the object (readonly, default 0).</li> <li>• -y2 The second y coordinate of the object (readonly, default 0).</li> <li>• -controlpoint The name of the control point this label is part of (readonly, default MainLine).</li> <li>• -label The label of the <a href="#">HiddenBlock</a> (default "").</li> <li>• -position The position of the label (readonly, default below).</li> <li>• -orientation The orientation of the bridge (8-way) (readonly, default 0).</li> <li>• -flipped Whether the bridge is flipped (readonly, default no).</li> <li>• -occupiedcommand A command to run to find out if the <a href="#">HiddenBlock</a> is occupied (default {}).</li> </ul>

Defined coords terminals:

- E1 First endpoint.
- E2 Second endpoint.

Defined values (states): none. Defined indicators: none.

**Author**

Robert Heller <heller@deepsoft.com>

**9.66.2 Constructor & Destructor Documentation****9.66.2.1 HiddenBlock()**

```
CTCPanel::HiddenBlock::HiddenBlock (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a [HiddenBlock](#) object.

**Parameters**

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the <a href="#">HiddenBlock</a> on.
<code>...</code>	Option list.

**9.66.2.2 ~HiddenBlock()**

```
CTCPanel::HiddenBlock::~~HiddenBlock ( )
```

Clean up all data objects and free up all resources.

**9.66.3 Member Function Documentation****9.66.3.1 \_configureLabel()**

```
CTCPanel::HiddenBlock::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

### 9.66.3.2 `geti()`

```
CTCPanel::HiddenBlock::geti (
    ind )
```

Method to get the state of one of our indicators (none).

### 9.66.3.3 `getv()`

```
CTCPanel::HiddenBlock::getv ( )
```

Method to get our value (state).

### 9.66.3.4 `invoke()`

```
CTCPanel::HiddenBlock::invoke ( )
```

Method to invoke the [HiddenBlock](#).

### 9.66.3.5 `seti()`

```
CTCPanel::HiddenBlock::seti (
    ind ,
    value )
```

Method to set an indicator's state (none).

### 9.66.3.6 `setv()`

```
CTCPanel::HiddenBlock::setv (
    value )
```

Method to set out value (state).

#### Parameters

<i>value</i>	The new state to set.
--------------	-----------------------

### 9.66.4 Member Data Documentation

#### 9.66.4.1 canvas

CTCPanel::HiddenBlock::canvas [private]

The canvas component (parent widget component).

#### 9.66.4.2 ctcpnl

CTCPanel::HiddenBlock::ctcpnl [private]

The CTC Panel component (parent widget).

## 9.67 HTMLHelp::HTMLHelp Class Reference

A widget that implements a help dialog that renders HTML coded help pages (generally generated from LaTeX using tex4ht's htlatex script).

### Public Member Functions

- [HTMLHelp](#) (name,...)  
*HTMLHelp constructor method.*
- [helpTopic](#) (topic="Help")  
*Public method to display help on a specific topic.*

### Static Public Member Functions

- static [GetInstance](#) (widget)  
*Returns the parent object given the specified child widget.*
- static [setDefault](#) (helpdir, [toc](#))  
*A public typemethod to set the default values for the -helpdirectory and -tableofcontents options.*
- static [help](#) (topic)  
*A public typemethod to create and launch a default help dialog.*

## Private Member Functions

- [\\_Close](#) ()
- [back](#) ()
- [forward](#) ()
- [nextlink](#) (w)
- [prevlink](#) (w)
- [searchforward](#) (w)
- [\\_SForward](#) (w)
- [searchbackward](#) (w)
- [\\_SBackward](#) (w)

## Static Private Member Functions

- static [findtopicintoc](#) (selfns, topic)
- static [pushcurrenttopic](#) (selfns, url)
- static [backcurrenttopic](#) (selfns)
- static [forwardcurrenttopic](#) (selfns)
- static [render](#) (selfns, win, url, push="yes")
- static [HMininit\\_win](#) (selfns, win)
- static [HMset\\_indent](#) (win, cm)
 

*set the indent spacing (in cm) for lists TK uses a "weird" tabbing model that causes to insert a single space if the current line position is past the tab setting*
- static [HMreset\\_win](#) (win)
 

*reset the state of window - get ready for the next page remove all but the font tags, and remove all form state*
- static [HMininit\\_state](#) (win)
 

*initialize the window's state array Parameters beginning with S\_ are NOT reset adjust\_size: global font size adjuster unknown: character to use for unknown entities tab: tab stop (in cm) stop: enabled to stop processing update: how many tags between update calls tags: number of tags processed so far symbols: Symbols to use on un-ordered lists*
- static [HMset\\_state](#) (win,...)
- static [HMrender](#) (selfns, win, tag, not, param, text)
 

*HMrender gets called for every html tag win: The name of the text widget to render into tag: The html tag (in arbitrary case) not: a "/" or the empty string param: The un-interpreted parameter list text: The plain text until the next html tag.*
- static [HMtag\\_hmstart](#) (selfns, win, param, text)
 

*A pair of pseudo tags are added automatically as the 1st and last html tags in the document.*
- static [HMtag\\_hmstart](#) (selfns, win, param, text)
- static [HMtag\\_title](#) (selfns, win, param, text)
 

*put the document title in the window banner, and remove the title text from the document*
- static [HMtag\\_hr](#) (selfns, win, param, text)
- static [HMtag\\_ol](#) (selfns, win, param, text)
- static [HMtag\\_ul](#) (selfns, win, param, text)
- static [HMtag\\_menu](#) (selfns, win, param, text)
- static [HMtag\\_menu](#) (selfns, win, param, text)
- static [HMtag\\_dt](#) (selfns, win, param, text)
- static [HMtag\\_li](#) (selfns, win, param, text)
- static [HMtag\\_a](#) (selfns, win, param, text)
 

*Manage hypertext "anchor" links.*
- static [HMgoto](#) (selfns, win, where, callback="HMwent\_to")
 

*The application should call here with the fragment name to cause the display to go to this spot.*

- static [HMwent\\_to](#) (selfns, win, where, count=0, color="orange")  
*We actually got to the spot, so highlight it! This should/could be replaced by the application We'll flash it orange a couple of times.*
- static [HMtag\\_a](#) (selfns, win, param, text)
- static [HMtag\\_img](#) (selfns, win, param, text)  
*Inline Images This interface is subject to change Most of the work is getting around a limitation of TK that prevents setting the size of a label to a widthxheight in pixels.*
- static [HMgot\\_image](#) (win, image\_error)  
*When the image is available, the application should call back here.*
- static [HMLink\\_setup](#) (win, href)  
*We need to escape any 's in the href tag name so the bind command doesn't try to substitute them.*
- static [HMLink\\_hit](#) (selfns, win, x, y)  
*generic link-hit callback This gets called upon button hits on hypertext links Applications are expected to supply their own HMLink\_callback routine win: The name of the text widget to render into x,y: The cursor position at the "click"*
- static [HMcheck\\_tocRelative](#) (link, tocfiler)
- static [HMextract\\_param](#) (param, key, val="")  
*extract a value from parameter list (this needs a re-do) returns "1" if the keyword is found, "0" otherwise param: A parameter list.*
- static [HMstack](#) (win, push, list)  
*Push or pop tags to/from stack.*
- static [HMcurrent\\_tags](#) (selfns, win)  
*extract set of current text tags tags starting with T map directly to text tags, all others are handled specially.*
- static [HMx\\_font](#) (family, size, weight, style, adjust\_size=0)  
*generate an X font name*
- static [HMOptimize](#) ()  
*Optimize HMrender (hee hee) This is experimental.*
- static [HMparse\\_html](#) (html, cmd="HMtest\_parse", start="hmstart")  
*Turn HTML into TCL commands html A string containing an html document cmd A command to run for each html tag found start The name of the dummy html start/stop tags.*
- static [HMtest\\_parse](#) (command, tag, slash, text\_after\_tag)
- static [HMzap\\_white](#) (data)  
*Convert multiple white space into a single space.*
- static [HMmap\\_esc](#) (text)  
*find HTML escape characters of the form*
- static [HMdo\\_map](#) (text, unknown="?")  
*convert an HTML escape sequence into character*
- static [HMtag\\_isindex](#) (selfns, win, param, text)  
*html isindex tag.*
- static [HMsubmit\\_index](#) (selfns, win, param, text)  
*This is called when the isindex form is submitted.*
- static [HMtag\\_form](#) (selfns, win, param, text)  
*initialize form state.*
- static [HMtag\\_form](#) (selfns, win, param, text)  
*Where we're done try to get all of the state into the widgets so we can free up the form structure here.*
- static [HMtag\\_input](#) (selfns, win, param, text)  
*handle form input items each item type is handled in a separate procedure Each "type" procedure needs to:*
- static [HMinput\\_text](#) (win, param, show="")  
*input type=text parameters NAME (reqd), MAXLENGTH, SIZE, VALUE*

- static [HMinput\\_password](#) (win, param)  
*password fields - same as text, only don't show data parameters NAME (reqd), MAXLENGTH, SIZE, VALUE*
- static [HMinput\\_checkbox](#) (win, param)  
*checkboxes are missing a "get" option, so we must use a global variable to store the value.*
- static [HMinput\\_radio](#) (win, param)  
*radio buttons.*
- static [HMinput\\_hidden](#) (win, param)  
*hidden fields, just append to the "submit" data params: NAME, VALUE (reqd)*
- static [HMinput\\_image](#) (win, param)  
*handle input images.*
- static [HMinput\\_reset](#) (win, param)  
*Set up the reset button.*
- static [HMinput\\_submit](#) (win, param)  
*Set up the submit button.*
- static [HMTag\\_select](#) (selfns, win, param, text)  
*selection items They all go into a list box.*
- static [HMTag\\_option](#) (selfns, win, param, text)  
*select options The values returned in the query may be different from those displayed in the listbox, so we need to keep a separate list of query values.*
- static [HMTag\\_\\_select](#) (selfns, win, param, text)  
*do most of the work here! if SIZE > 1, make the listbox.*
- static [HMTag\\_textarea](#) (selfns, win, param, text)  
*do a text area (multi-line text) params: COLS, NAME, ROWS (all reqd, but default rows and cols anyway)*
- static [HMwin\\_install](#) (win, item)  
*procedure to install windows into the text widget*
- static [HMsubmit\\_button](#) (win, form\_id, param, stuff)  
*Assemble and submit the query each list element in "stuff" is a name/value pair.*
- static [HMsubmit\\_form](#) (win, param, query)  
*sample user callback for form submission should be replaced by the application Sample version generates a string suitable for http*
- static [HMmap\\_reply](#) (string)  
*1 leave alphanumerics characters alone 2 Convert every other character to an array lookup 3 Escape constructs that are "special" to the tcl parser 4 "subst" the result, doing all the array substitutions*
- static [HMcgiDecode](#) (data)  
*convert a x-www-urlencoded string into a list of name/value pairs 1 convert a=b&c=d...*
- static [HMcgiMap](#) (data)
- static [get\\_html](#) (file)  
*given a file name, return its html, or invent some html if the file can't be opened.*
- static [HMLink\\_callback](#) (selfns, win, href)  
*Override the library link-callback routine for the sample app.*
- static [HMset\\_image](#) (selfns, win, handle, src)  
*Supply an image callback function Read in an image if we don't already have one callback to library for display.*
- static [HMset\\_font](#) (selfns, win, tag, font)  
*downloading fonts can take a long time.*
- static [HMTag\\_color](#) (selfns, win, param, text)  
*Lets invent a new HTML tag, just for fun.*
- static [HMTag\\_\\_color](#) (selfns, win, param, text)
- static [HMTag\\_font](#) (selfns, win, param, text)



*Add a font size manipulation primitive, so we can use this sample program for on-line presentations.*

- static [HMTag\\_font](#) (selfns, win, param, text)

*This version is closer to what Netscape does.*

- static [HMTag\\_font](#) (selfns, win, param, text)
- static [HMTag\\_link](#) (selfns, win, param, text)
- static [HMload\\_css](#) (selfns, win, href)
- static [HMappend\\_css](#) (varName, cssBlock)

## Private Attributes

- [panes](#)

*PaneWindow component.*

- [tocscroll](#)

*ScrolledWindow for the Table Of Contents component.*

- [toc](#)

*The Table Of Contents component.*

- [toc\\_css](#)

*The Table Of Contents stylesheet.*

- [textscroll](#)

*The text area ScrolledWindow component.*

- [helptext](#)

*The text area component.*

- [helptext\\_css](#)

*The text area stylesheet.*

- [status](#)

*The Help status component.*

- [command](#)

*The Help command component.*

- [Url](#)

*The current URL.*

- [topicstack](#)

- [curtopicindex](#)

- [lastsearch](#)

*initialize the window and stack state*

- [Fonts](#)

## Static Private Attributes

- static [\\_WidgetMap](#)

*Widget map.*

- static [defaultHelpDirectory](#)

*The default help directory.*

- static [defaultTableOfContents](#)

*The default Table Of Contents file.*

- static [defaultHelpWindow](#)

*The default Help file.*

- static [HMTag\\_map](#)
- static [HMinert\\_map](#)
- static [HMList\\_elements](#)
- static [HMparam\\_map](#)
- static [HMevents](#)
- static [HMform\\_map](#)
- static [HMesc\\_map](#)
- static [HMalphanumeric](#)

### 9.67.1 Detailed Description

A widget that implements a help dialog that renders HTML coded help pages (generally generated from LaTeX using tex4ht's htlatex script).

This widget contains two panes, a narrow contents (a link menu) pane on the left and a wider help text pane on the right. Below these panes is a status label and a search text labeled entry, with a set of four buttons along the bottom.

#### Parameters

<i>path</i>	Pathname of the widget.
...	Options: <ul style="list-style-type: none"> <li>• -textwidth The initial width of the text pane.</li> <li>• -width The overall width of the help dialog.</li> <li>• -height The overall height of the help dialog.</li> <li>• -side The side, top or bottom to put the sash. Default top.</li> <li>• -helpdirectory The directory where the HTML files reside.</li> <li>• -tableofcontents The file in the help directory that contains the table of contents (link menu).</li> </ul>

#### Author

Stephen Uhler <stephen.uhler@sun.com>, Clif Flynt <clif@cflynt.com>, and Robert Heller <heller@deepsoft.com>.

### 9.67.2 Constructor & Destructor Documentation

#### 9.67.2.1 HTMLHelp()

```
HTMLHelp::HTMLHelp::HTMLHelp (
    name ,
    ... )
```

[HTMLHelp](#) constructor method.

The [HTMLHelp](#) is constructed here.

#### Parameters

...	Option value pairs.
-----	---------------------

### 9.67.3 Member Function Documentation

#### 9.67.3.1 \_Close()

```
HTMLHelp::HTMLHelp::_Close ( ) [private]
```

#### 9.67.3.2 \_SBackward()

```
HTMLHelp::HTMLHelp::_SBackward (
    w ) [private]
```

#### 9.67.3.3 \_SForward()

```
HTMLHelp::HTMLHelp::_SForward (
    w ) [private]
```

#### 9.67.3.4 a()

```
static HMTAG_ HTMLHelp::HTMLHelp::a (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

#### 9.67.3.5 back()

```
HTMLHelp::HTMLHelp::back ( ) [private]
```

#### 9.67.3.6 backcurrenttopic()

```
static HTMLHelp::HTMLHelp::backcurrenttopic (
    selfns ) [static], [private]
```

#### 9.67.3.7 color()

```
static HMTAG_HTMLHelp::HTMLHelp::color (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

#### 9.67.3.8 findtopicintoc()

```
static HTMLHelp::HTMLHelp::findtopicintoc (
    selfns ,
    topic ) [static], [private]
```

#### 9.67.3.9 font()

```
static HMTAG_HTMLHelp::HTMLHelp::font (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

#### 9.67.3.10 form()

```
static HMTAG_HTMLHelp::HTMLHelp::form (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

Where we're done try to get all of the state into the widgets so we can free up the form structure here.

Unfortunately, we can't!

#### 9.67.3.11 forward()

```
HTMLHelp::HTMLHelp::forward ( ) [private]
```

#### 9.67.3.12 forwardcurrenttopic()

```
static HTMLHelp::HTMLHelp::forwardcurrenttopic (
    selfns ) [static], [private]
```

#### 9.67.3.13 get\_html()

```
static HTMLHelp::HTMLHelp::get_html (
    file ) [static], [private]
```

given a file name, return its html, or invent some html if the file can't be opened.

#### 9.67.3.14 GetInstance()

```
static HTMLHelp::HTMLHelp::GetInstance (
    widget ) [static]
```

Returns the parent object given the specified child widget.

#### 9.67.3.15 help()

```
static HTMLHelp::HTMLHelp::help (
    topic ) [static]
```

A public typemethod to create and launch a default help dialog.

The setDefaults typemethod must be called before this typemethod!

##### Parameters

<i>topic</i>	The help topic text to display help for.
--------------	--

#### 9.67.3.16 helpTopic()

```
HTMLHelp::HTMLHelp::helpTopic (  
    topic = "Help" )
```

Public method to display help on a specific topic.

##### Parameters

<i>topic</i>	The topic text to display help for.
--------------	-------------------------------------

#### 9.67.3.17 HMappend\_css()

```
static HTMLHelp::HTMLHelp::HMappend_css (  
    varName ,  
    cssBlock ) [static], [private]
```

#### 9.67.3.18 HMcgiDecode()

```
static HTMLHelp::HTMLHelp::HMcgiDecode (  
    data ) [static], [private]
```

convert a x-www-urlencoded string into a list of name/value pairs 1 convert a=b&c=d...

to {a} {b} {c} {d}... 2, convert + to " " 3, convert xx to char equiv

#### 9.67.3.19 HMcgiMap()

```
static HTMLHelp::HTMLHelp::HMcgiMap (  
    data ) [static], [private]
```

#### 9.67.3.20 HMcheck\_tocRelative()

```
static HTMLHelp::HTMLHelp::HMcheck_tocRelative (  
    link ,  
    tocfile ) [static], [private]
```

#### 9.67.3.21 HMcurrent\_tags()

```
static HTMLHelp::HTMLHelp::HMcurrent_tags (
    selfns ,
    win ) [static], [private]
```

extract set of current text tags tags starting with T map directly to text tags, all others are handled specially.

There is an application callback, HMset\_font to allow the application to do font error handling

#### 9.67.3.22 HMdo\_map()

```
static HTMLHelp::HTMLHelp::HMdo_map (
    text ,
    unknown = "?" ) [static], [private]
```

convert an HTML escape sequence into character

#### 9.67.3.23 HMextract\_param()

```
static HTMLHelp::HTMLHelp::HMextract_param (
    param ,
    key ,
    val = "" ) [static], [private]
```

extract a value from parameter list (this needs a re-do) returns "1" if the keyword is found, "0" otherwise param: A parameter list.

It should already have been processed to remove any entity references key: The parameter name val: The variable to put the value into (use key as default)

#### 9.67.3.24 HMgot\_image()

```
static HTMLHelp::HTMLHelp::HMgot_image (
    win ,
    image_error ) [static], [private]
```

When the image is available, the application should call back here.

If we have the image, put it in the label, otherwise display the error message. If we don't get a callback, the "alt" text remains. if we have a clickable image, arrange for a callback

**9.67.3.25 HMgoto()**

```
static HTMLHelp::HTMLHelp::HMgoto (
    selfns ,
    win ,
    where ,
    callback = "HMwent_to" ) [static], [private]
```

The application should call here with the fragment name to cause the display to go to this spot.

If the target exists, go there (and do the callback), otherwise schedule the goto to happen when we see the reference.

**9.67.3.26 HMininit\_state()**

```
static HTMLHelp::HTMLHelp::HMininit_state (
    win ) [static], [private]
```

initialize the window's state array Parameters beginning with S\_ are NOT reset adjust\_size: global font size adjuster  
unknown: character to use for unknown entities tab: tab stop (in cm) stop: enabled to stop processing update: how  
many tags between update calls tags: number of tags processed so far symbols: Symbols to use on un-ordered lists

**9.67.3.27 HMininit\_win()**

```
static HTMLHelp::HTMLHelp::HMininit_win (
    selfns ,
    win ) [static], [private]
```

**9.67.3.28 HMininput\_checkbox()**

```
static HTMLHelp::HTMLHelp::HMininput_checkbox (
    win ,
    param ) [static], [private]
```

checkboxbuttons are missing a "get" option, so we must use a global variable to store the value.

Parameters NAME, VALUE, (reqd), CHECKED

**9.67.3.29 HMininput\_hidden()**

```
static HTMLHelp::HTMLHelp::HMininput_hidden (
    win ,
    param ) [static], [private]
```

hidden fields, just append to the "submit" data params: NAME, VALUE (reqd)



### 9.67.3.30 HMinput\_image()

```
static HTMLHelp::HTMLHelp::HMinput_image (  
    win ,  
    param ) [static], [private]
```

handle input images.

The spec isn't very clear on these, so I'm not sure its quite right Use std image tag, only set up our own callbacks (e.g. make sure ismap isn't set) params: NAME, SRC (reqd) ALIGN

### 9.67.3.31 HMinput\_password()

```
static HTMLHelp::HTMLHelp::HMinput_password (  
    win ,  
    param ) [static], [private]
```

password fields - same as text, only don't show data parameters NAME (reqd), MAXLENGTH, SIZE, VALUE

### 9.67.3.32 HMinput\_radio()

```
static HTMLHelp::HTMLHelp::HMinput_radio (  
    win ,  
    param ) [static], [private]
```

radio buttons.

These are like check buttons, but only one can be selected

### 9.67.3.33 HMinput\_reset()

```
static HTMLHelp::HTMLHelp::HMinput_reset (  
    win ,  
    param ) [static], [private]
```

Set up the reset button.

Wait for the /form to attach the -command option. There could be more that 1 reset button params VALUE

### 9.67.3.34 HMinput\_submit()

```
static HTMLHelp::HTMLHelp::HMinput_submit (  
    win ,  
    param ) [static], [private]
```

Set up the submit button.

Wait for the /form to attach the -command option. There could be more that 1 submit button params: NAME, VALUE

#### 9.67.3.35 HMinput\_text()

```
static HTMLHelp::HTMLHelp::HMinput_text (
    win ,
    param ,
    show = "" ) [static], [private]
```

input type=text parameters NAME (reqd), MAXLENGTH, SIZE, VALUE

#### 9.67.3.36 Hmlink\_callback()

```
static HTMLHelp::HTMLHelp::Hmlink_callback (
    selfns ,
    win ,
    href ) [static], [private]
```

Override the library link-callback routine for the sample app.

It only handles the simple cases.

#### 9.67.3.37 Hmlink\_hit()

```
static HTMLHelp::HTMLHelp::Hmlink_hit (
    selfns ,
    win ,
    x ,
    y ) [static], [private]
```

generic link-hit callback This gets called upon button hits on hypertext links Applications are expected to supply their own Hmlink\_callback routine win: The name of the text widget to render into x,y: The cursor position at the "click"

#### 9.67.3.38 Hmlink\_setup()

```
static HTMLHelp::HTMLHelp::Hmlink_setup (
    win ,
    href ) [static], [private]
```

We need to escape any 's in the href tag name so the bind command doesn't try to substitute them.

#### 9.67.3.39 HMload\_css()

```
static HTMLHelp::HTMLHelp::HMload_css (
    selfns ,
    win ,
    href ) [static], [private]
```

#### 9.67.3.40 HMmap\_esc()

```
static HTMLHelp::HTMLHelp::HMmap_esc (
    text ) [static], [private]
```

find HTML escape characters of the form

#### 9.67.3.41 HMmap\_reply()

```
static HTMLHelp::HTMLHelp::HMmap_reply (
    string ) [static], [private]
```

1 leave alphanumerics characters alone 2 Convert every other character to an array lookup 3 Escape constructs that are "special" to the tcl parser 4 "subst" the result, doing all the array substitutions

#### 9.67.3.42 HMOptimize()

```
static HTMLHelp::HTMLHelp::HMOptimize ( ) [static], [private]
```

Optimize HMrender (hee hee) This is experimental.

#### 9.67.3.43 HMparse\_html()

```
static HTMLHelp::HTMLHelp::HMparse_html (
    html ,
    cmd = "HMtest_parse",
    start = "hmstart" ) [static], [private]
```

Turn HTML into TCL commands html A string containing an html document cmd A command to run for each html tag found start The name of the dummy html start/stop tags.

#### 9.67.3.44 HMrender()

```
static HTMLHelp::HTMLHelp::HMrender (
    selfns ,
    win ,
    tag ,
    not ,
    param ,
    text ) [static], [private]
```

HMrender gets called for every html tag win: The name of the text widget to render into tag: The html tag (in arbitrary case) not: a "/" or the empty string param: The un-interpreted parameter list text: The plain text until the next html tag.

#### 9.67.3.45 HMreset\_win()

```
static HTMLHelp::HTMLHelp::HMreset_win (
    win ) [static], [private]
```

reset the state of window - get ready for the next page remove all but the font tags, and remove all form state

#### 9.67.3.46 HMset\_font()

```
static HTMLHelp::HTMLHelp::HMset_font (
    selfns ,
    win ,
    tag ,
    font ) [static], [private]
```

downloading fonts can take a long time.

We'll override the default font-setting routine to permit better user feedback on fonts. We'll keep our own list of installed fonts on the side, to guess when delays are likely

#### 9.67.3.47 HMset\_image()

```
static HTMLHelp::HTMLHelp::HMset_image (
    selfns ,
    win ,
    handle ,
    src ) [static], [private]
```

Supply an image callback function Read in an image if we don't already have one callback to library for display.

#### 9.67.3.48 HMset\_indent()

```
static HTMLHelp::HTMLHelp::HMset_indent (
    win ,
    cm ) [static], [private]
```

set the indent spacing (in cm) for lists TK uses a "weird" tabbing model that causes to insert a single space if the current line position is past the tab setting

#### 9.67.3.49 HMset\_state()

```
static HTMLHelp::HTMLHelp::HMset_state (
    win ,
    ... ) [static], [private]
```

#### 9.67.3.50 HMstack()

```
static HTMLHelp::HTMLHelp::HMstack (
    win ,
    push ,
    list ) [static], [private]
```

Push or pop tags to/from stack.

Each orthogonal text property has its own stack, stored as a list. The current (most recent) tag is the last item on the list. Push is {} for pushing and {} for popping

#### 9.67.3.51 hmstart()

```
static HMTAG_ HTMLHelp::HTMLHelp::hmstart (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

#### 9.67.3.52 HSubmit\_button()

```
static HTMLHelp::HTMLHelp::HSubmit_button (
    win ,
    form_id ,
    param ,
    stuff ) [static], [private]
```

Assemble and submit the query each list element in "stuff" is a name/value pair.

- The names are the NAME parameters of the various fields
- The values get run through "subst" to extract the values
- We do the user callback with the list of name value pairs

#### 9.67.3.53 HSubmit\_form()

```
static HTMLHelp::HTMLHelp::HSubmit_form (
    win ,
    param ,
    query ) [static], [private]
```

sample user callback for form submission should be replaced by the application Sample version generates a string suitable for http

#### 9.67.3.54 HSubmit\_index()

```
static HTMLHelp::HTMLHelp::HSubmit_index (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

This is called when the isindex form is submitted.

The default version calls Hmlink\_callback. Isindex tags should either be deprecated, or fully supported (e.g. they need an href parameter)

#### 9.67.3.55 HMTag\_a()

```
static HTMLHelp::HTMLHelp::HMTag_a (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

Manage hypertext "anchor" links.

A link can be either a source (href) a destination (name) or both. If its a source, register it via a callback, and set its default behavior. If its a destination, check to see if we need to go there now, as a result of a previous HMgoto request. If so, schedule it to happen with the closing </a> tag, so we can highlight the text up to the </a>.

#### 9.67.3.56 HMTag\_color()

```
static HTMLHelp::HTMLHelp::HMTag_color (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

Lets invent a new HTML tag, just for fun.

Change the color of the text. Use html tags of the form: <color value="blue"> ... </color> We can invent a new tag for the display stack. If it starts with "T" it will automatically get mapped directly to a text widget tag.

#### 9.67.3.57 HMTag\_dt()

```
static HTMLHelp::HTMLHelp::HMTag_dt (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

#### 9.67.3.58 HMTag\_font() [1/2]

```
static HTMLHelp::HTMLHelp::HMTag_font (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

Add a font size manipulation primitive, so we can use this sample program for on-line presentations.

sizes prefixed with + or - are relative. <font size=[+]-3> ..... </font>. Note that this is not the same as Netscape's <font> tag.

#### 9.67.3.59 HMTag\_font() [2/2]

```
static HTMLHelp::HTMLHelp::HMTag_font (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

This version is closer to what Netscape does.

#### 9.67.3.60 HMTag\_form()

```
static HTMLHelp::HTMLHelp::HMTag_form (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

initialize form state.

All of the state for this form is kept in a global array whose name is stored in the form\_id field of the main window array.  
Parameters: ACTION, METHOD, ENCTYPE

#### 9.67.3.61 HMTag\_hmstart()

```
static HTMLHelp::HTMLHelp::HMTag_hmstart (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

A pair of pseudo tags are added automatically as the 1st and last html tags in the document.

The default is <HMstart> and </HMstart>. Append enough blank space at the end of the text widget while rendering so HMgoto can place the target near the top of the page, then remove the extra space when done rendering.

#### 9.67.3.62 HMTag\_hr()

```
static HTMLHelp::HTMLHelp::HMTag_hr (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```



### 9.67.3.63 HMTag\_img()

```
static HTMLHelp::HTMLHelp::HMTag_img (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

Inline Images This interface is subject to change Most of the work is getting around a limitation of TK that prevents setting the size of a label to a widthxheight in pixels.

Images have the following parameters: align: top,middle,bottom alt: alternate text ismap: A clickable image map src: The URL link Netscape supports (and so do we) width: A width hint (in pixels) height: A height hint (in pixels) border: The size of the window border

### 9.67.3.64 HMTag\_input()

```
static HTMLHelp::HTMLHelp::HMTag_input (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

handle form input items each item type is handled in a separate procedure Each "type" procedure needs to:

- create the window
- initialize it
- add the "submit" and "reset" commands onto the proper Q's "submit" is subst'd "reset" is eval'd

### 9.67.3.65 HMTag\_isindex()

```
static HTMLHelp::HTMLHelp::HMTag_isindex (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

html isindex tag.

Although not strictly forms, they're close enough to be in this file is-index forms make a frame with a label, entry, and submit button

**9.67.3.66 HMTag\_li()**

```
static HTMLHelp::HTMLHelp::HMTag_li (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

**9.67.3.67 HMTag\_link()**

```
static HTMLHelp::HTMLHelp::HMTag_link (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

**9.67.3.68 HMTag\_menu()**

```
static HTMLHelp::HTMLHelp::HMTag_menu (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

**9.67.3.69 HMTag\_ol()**

```
static HTMLHelp::HTMLHelp::HMTag_ol (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

**9.67.3.70 HMTag\_option()**

```
static HTMLHelp::HTMLHelp::HMTag_option (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

**select options** The values returned in the query may be different from those displayed in the listbox, so we need to keep a separate list of query values.

**form(select\_default)** - contains the default query value  
**form(select\_frame)** - name of the listbox's containing frame  
**form(select\_values)** - list of query values  
params: VALUE, SELECTED

#### 9.67.3.71 HMTag\_select()

```
static HTMLHelp::HTMLHelp::HMTag_select (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

selection items They all go into a list box.

We don't what to do with the listbox until we know how many items end up in it. Gather up the data for the "options" and finish up in the /select tag params: NAME (reqd), MULTIPLE, SIZE

#### 9.67.3.72 HMTag\_textarea()

```
static HTMLHelp::HTMLHelp::HMTag_textarea (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

do a text area (multi-line text) params: COLS, NAME, ROWS (all reqd, but default rows and cols anyway)

#### 9.67.3.73 HMTag\_title()

```
static HTMLHelp::HTMLHelp::HMTag_title (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

put the document title in the window banner, and remove the title text from the document

#### 9.67.3.74 HMTag\_ul()

```
static HTMLHelp::HTMLHelp::HMTag_ul (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

#### 9.67.3.75 HMtest\_parse()

```
static HTMLHelp::HTMLHelp::HMtest_parse (
    command ,
    tag ,
    slash ,
    text_after_tag ) [static], [private]
```

#### 9.67.3.76 HMwent\_to()

```
static HTMLHelp::HTMLHelp::HMwent_to (
    selfns ,
    win ,
    where ,
    count = 0,
    color = "orange" ) [static], [private]
```

We actually got to the spot, so highlight it! This should/could be replaced by the application We'll flash it orange a couple of times.

#### 9.67.3.77 HMwin\_install()

```
static HTMLHelp::HTMLHelp::HMwin_install (
    win ,
    item ) [static], [private]
```

procedure to install windows into the text widget

- win: name of the text widget
- item: name of widget to install

#### 9.67.3.78 HMx\_font()

```
static HTMLHelp::HTMLHelp::HMx_font (
    family ,
    size ,
    weight ,
    style ,
    adjust_size = 0 ) [static], [private]
```

generate an X font name

**9.67.3.79 HMzap\_white()**

```
static HTMLHelp::HTMLHelp::HMzap_white (  
    data ) [static], [private]
```

Convert multiple white space into a single space.

**9.67.3.80 menu()**

```
static HMTAG_ HTMLHelp::HTMLHelp::menu (  
    selfns ,  
    win ,  
    param ,  
    text ) [static], [private]
```

**9.67.3.81 nextlink()**

```
HTMLHelp::HTMLHelp::nextlink (  
    w ) [private]
```

**9.67.3.82 prevlink()**

```
HTMLHelp::HTMLHelp::prevlink (  
    w ) [private]
```

**9.67.3.83 pushcurrenttopic()**

```
static HTMLHelp::HTMLHelp::pushcurrenttopic (  
    selfns ,  
    url ) [static], [private]
```

#### 9.67.3.84 render()

```
static HTMLHelp::HTMLHelp::render (
    selfns ,
    win ,
    url ,
    push = "yes" ) [static], [private]
```

#### 9.67.3.85 searchbackward()

```
HTMLHelp::HTMLHelp::searchbackward (
    w ) [private]
```

#### 9.67.3.86 searchforward()

```
HTMLHelp::HTMLHelp::searchforward (
    w ) [private]
```

#### 9.67.3.87 select()

```
static HMTAG_ HTMLHelp::HTMLHelp::select (
    selfns ,
    win ,
    param ,
    text ) [static], [private]
```

do most of the work here! if SIZE>1, make the listbox.

Otherwise make a "drop-down" listbox with a label in it If the # of items > size, add a scroll bar This should probably be broken up into callbacks to make it easier to override the "look".

#### 9.67.3.88 setDefaults()

```
static HTMLHelp::HTMLHelp::setDefaults (
    helpdir ,
    toc ) [static]
```

A public typemethod to set the default values for the -helpdirectory and -tableofcontents options.

## Parameters

<i>helpdir</i>	The default value for -helpdirectory.
<i>toc</i>	The default value for -tableofcontents.

## 9.67.4 Member Data Documentation

### 9.67.4.1 \_WidgetMap

HTMLHelp::HTMLHelp::\_WidgetMap [static], [private]

Widget map.

### 9.67.4.2 command

HTMLHelp::HTMLHelp::command [private]

The Help command component.

### 9.67.4.3 curtopicindex

HTMLHelp::HTMLHelp::curtopicindex [private]

### 9.67.4.4 defaultHelpDirectory

HTMLHelp::HTMLHelp::defaultHelpDirectory [static], [private]

The default help directory.

#### 9.67.4.5 defaultHelpWindow

HTMLHelp::HTMLHelp::defaultHelpWindow [static], [private]

The default Help file.

#### 9.67.4.6 defaultTableOfContents

HTMLHelp::HTMLHelp::defaultTableOfContents [static], [private]

The default Table Of Contents file.

#### 9.67.4.7 Fonts

HTMLHelp::HTMLHelp::Fonts [private]

#### 9.67.4.8 helptext

HTMLHelp::HTMLHelp::helptext [private]

The text area component.

#### 9.67.4.9 helptext\_css

HTMLHelp::HTMLHelp::helptext\_css [private]

The text area stylesheet.

#### 9.67.4.10 HMalphanumeric

HTMLHelp::HTMLHelp::HMalphanumeric [static], [private]



#### 9.67.4.11 HMesc\_map

HTMLHelp::HTMLHelp::HMesc\_map [static], [private]

#### 9.67.4.12 HMevents

HTMLHelp::HTMLHelp::HMevents [static], [private]

#### 9.67.4.13 HMform\_map

HTMLHelp::HTMLHelp::HMform\_map [static], [private]

#### 9.67.4.14 HMinert\_map

HTMLHelp::HTMLHelp::HMinert\_map [static], [private]

#### 9.67.4.15 HMlist\_elements

HTMLHelp::HTMLHelp::HMlist\_elements [static], [private]

#### 9.67.4.16 HMparam\_map

HTMLHelp::HTMLHelp::HMparam\_map [static], [private]

#### 9.67.4.17 HMtag\_map

HTMLHelp::HTMLHelp::HMtag\_map [static], [private]

#### 9.67.4.18 lastsearch

HTMLHelp::HTMLHelp::lastsearch [private]

initialize the window and stack state

html forms management commands

manage the display of html

#### 9.67.4.19 panes

HTMLHelp::HTMLHelp::panes [private]

PaneWindow component.

#### 9.67.4.20 status

HTMLHelp::HTMLHelp::status [private]

The Help status component.

#### 9.67.4.21 textscroll

HTMLHelp::HTMLHelp::textscroll [private]

The text area ScrolledWindow component.

#### 9.67.4.22 toc

HTMLHelp::HTMLHelp::toc [private]

The Table Of Contents component.

**9.67.4.23 toc\_css**

```
HTMLHelp::HTMLHelp::toc_css [private]
```

The Table Of Contents stylesheet.

**9.67.4.24 tocscroll**

```
HTMLHelp::HTMLHelp::tocscroll [private]
```

ScrolledWindow for the Table Of Contents component.

**9.67.4.25 topicstack**

```
HTMLHelp::HTMLHelp::topicstack [private]
```

**9.67.4.26 Url**

```
HTMLHelp::HTMLHelp::Url [private]
```

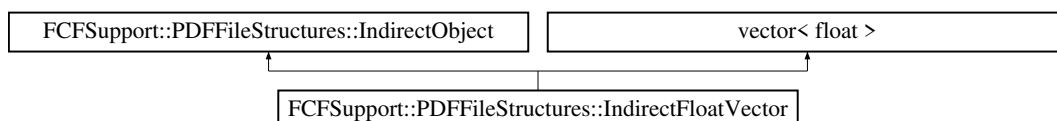
The current URL.

**9.68 FCFSupport::PDFFileStructures::IndirectFloatVector Class Reference**

Indirect array of floats.

```
#include <PDFPrinterSupport.h>
```

Inheritance diagram for FCFSupport::PDFFileStructures::IndirectFloatVector:



## Public Member Functions

- [IndirectFloatVector](#) (unsigned long int objNum=0L, unsigned short int genNum=0, CrossReferenceTable \*tab=↔ NULL)  
*Constructor.*
- [~IndirectFloatVector](#) ()  
*Destructor.*
- virtual ostream & [WriteDirect](#) (ostream &stream) const  
*Write an object directly.*

### 9.68.1 Detailed Description

Indirect array of floats.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.68.2 Constructor & Destructor Documentation

#### 9.68.2.1 IndirectFloatVector()

```
FCFSupport::PDFFileStructures::IndirectFloatVector::IndirectFloatVector (
    unsigned long int objNum = 0L,
    unsigned short int genNum = 0,
    CrossReferenceTable * tab = NULL ) [inline]
```

Constructor.

Create an indirect object of floats.

#### Parameters

<i>objNum</i>	The next object number.
<i>genNum</i>	The generation number.
<i>tab</i>	The cross reference table we are in.

#### 9.68.2.2 ~IndirectFloatVector()

```
FCFSupport::PDFFileStructures::IndirectFloatVector::~~IndirectFloatVector ( )
```

Destructor.



*Write an object directly.*

- unsigned long int [ObjectNumber](#) () const

*Return this object's object number.*

- unsigned short int [GenerationNumber](#) () const

*Return this object's generation number.*

- streampos [FileOffset](#) () const

*Return this object's file offset.*

- bool [HasOffset](#) () const

*Does the object have a file offset?*

## Private Member Functions

- void [SetObjectNumber](#) (unsigned long int on, [CrossReferenceTable](#) \*tab)

*Set this object's object number.*

- void [IncrementGenerationNumber](#) ()

*Increment the generation number.*

## Private Attributes

- unsigned long int [objectNumber](#)

*The object number.*

- unsigned short int [generationNumber](#)

*The generation number.*

- streampos [fileOffset](#)

*The file position.*

- [CrossReferenceTable](#) \* [table](#)

*The cross referece table we are in.*

## Friends

- class [CrossReferenceTable](#)

### 9.69.1 Detailed Description

Indirect object base class.

All PDF objects that might be referenced as indirect objects are derived from this class.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.69.2 Constructor & Destructor Documentation

### 9.69.2.1 IndirectObject()

```
FCFSupport::PDFFileStructures::IndirectObject::IndirectObject (
    unsigned long int objNum = 0L,
    unsigned short int genNum = 0,
    CrossReferenceTable * tab = NULL ) [inline]
```

Constructor.

Perform base initialization.

#### Parameters

<i>objNum</i>	The object number. Zero means no object number yet.
<i>genNum</i>	The generation number. Zero means first generation.
<i>tab</i>	The cross reference table we are in.

### 9.69.2.2 ~IndirectObject()

```
virtual FCFSupport::PDFFileStructures::IndirectObject::~~IndirectObject ( ) [inline], [virtual]
```

Destructor.

Clean everything up.

## 9.69.3 Member Function Documentation

### 9.69.3.1 FileOffset()

```
streampos FCFSupport::PDFFileStructures::IndirectObject::FileOffset ( ) const [inline]
```

Return this object's file offset.

### 9.69.3.2 GenerationNumber()

```
unsigned short int FCFSupport::PDFFileStructures::IndirectObject::GenerationNumber ( ) const [inline]
```

Return this object's generation number.

### 9.69.3.3 HasOffset()

```
bool FCFSupport::PDFFileStructures::IndirectObject::HasOffset ( ) const [inline]
```

Does the object have a file offset?

Referenced by FCFSupport::PDFFileStructures::IndirectObjectDictionary::WriteDictionaryElements().

### 9.69.3.4 IncrementGenerationNumber()

```
void FCFSupport::PDFFileStructures::IndirectObject::IncrementGenerationNumber ( ) [inline], [private]
```

Increment the generation number.

### 9.69.3.5 ObjectNumber()

```
unsigned long int FCFSupport::PDFFileStructures::IndirectObject::ObjectNumber ( ) const [inline]
```

Return this object's object number.

### 9.69.3.6 SetObjectNumber()

```
void FCFSupport::PDFFileStructures::IndirectObject::SetObjectNumber (
    unsigned long int on,
    CrossReferenceTable * tab ) [inline], [private]
```

Set this object's object number.

Should only be called when this object is inserted into a cross reference table. The object number cannot be reset!

#### Parameters

<i>on</i>	The object number to set this object to. Can't be zero. If the object number is already set, this can be the same number, in which case the generation number gets incremented.
<i>tab</i>	The cross reference table this object is being added to.



### 9.69.3.7 WriteDirect()

```
virtual ostream& FCFSupport::PDFFileStructures::IndirectObject::WriteDirect (
    ostream & stream ) const [pure virtual]
```

Write an object directly.

Needs to be overloaded by derived classes.

#### Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

Implemented in [FCFSupport::PDFFileStructures::IndirectFloatVector](#), [FCFSupport::PDFFileStructures::PDFStream](#), [FCFSupport::PDFFileStructures::Rectangle](#), [FCFSupport::PDFFileStructures::Dictionary](#), and [FCFSupport::PDFFileStructures::FreedObject](#).

Referenced by [FCFSupport::PDFFileStructures::IndirectObjectDictionary::WriteDictionaryElements\(\)](#).

### 9.69.3.8 WriteIndirectReference()

```
ostream& FCFSupport::PDFFileStructures::IndirectObject::WriteIndirectReference (
    ostream & stream ) const [inline]
```

Write an object indirectly.

Assumes that the non-const version has already been called. This version only writes an indirect reference.

#### Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

References `lcc::stream`.

Referenced by [FCFSupport::PDFFileStructures::IndirectObjectDictionary::WriteDictionaryElements\(\)](#).

### 9.69.3.9 WriteObjectToFile()

```
ostream& FCFSupport::PDFFileStructures::IndirectObject::WriteObjectToFile (
    ostream & stream ) [inline]
```

Write an object indirectly.

The first time this is called, an obj ... endobj block is written. Later times an indirect reference is written.

**Parameters**

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

References lcc::stream.

## 9.69.4 Friends And Related Function Documentation

### 9.69.4.1 CrossReferenceTable

```
friend class CrossReferenceTable [friend]
```

## 9.69.5 Member Data Documentation

### 9.69.5.1 fileOffset

```
streampos FCFSupport::PDFFileStructures::IndirectObject::fileOffset [private]
```

The file position.

### 9.69.5.2 generationNumber

```
unsigned short int FCFSupport::PDFFileStructures::IndirectObject::generationNumber [private]
```

The generation number.

### 9.69.5.3 objectNumber

```
unsigned long int FCFSupport::PDFFileStructures::IndirectObject::objectNumber [private]
```

The object number.

## 9.69.5.4 table

```
CrossReferenceTable* FCFSupport::PDFFileStructures::IndirectObject::table [private]
```

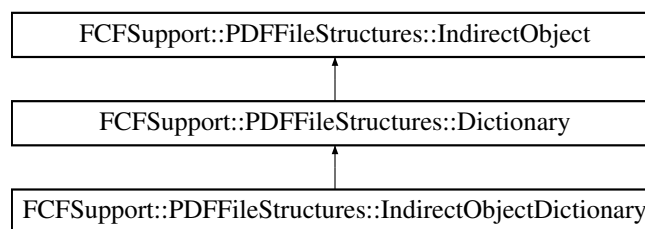
The cross referece table we are in.

## 9.70 FCFSupport::PDFFileStructures::IndirectObjectDictionary Class Reference

PDF Indirect Object [Dictionary](#), used for named resources in a Resource [Dictionary](#).

```
#include <PDFPrinterSupport.h>
```

Inheritance diagram for FCFSupport::PDFFileStructures::IndirectObjectDictionary:



### Public Member Functions

- [IndirectObjectDictionary](#) (unsigned long int objNum=0L, unsigned short int genNum=0, CrossReferenceTable \*tab=NULL)  
*Constructor.*
- [~IndirectObjectDictionary](#) ()  
*Destructor.*
- void [AddIndirectObject](#) (const string name, [IndirectObject](#) \*obj)  
*Add an indirect object.*
- int [Size](#) () const  
*Return the number of elements in this dictionary.*

### Protected Member Functions

- virtual ostream & [WriteDictionaryElements](#) (ostream &stream) const  
*Write the elements of a dictionary.*

### Private Attributes

- [NamedIndirectObjectMap](#) elements  
*The elements in this dictionary.*

### 9.70.1 Detailed Description

PDF Indirect Object [Dictionary](#), used for named resources in a Resource [Dictionary](#).

### 9.70.2 Constructor & Destructor Documentation

#### 9.70.2.1 IndirectObjectDictionary()

```
FCFSupport::PDFFileStructures::IndirectObjectDictionary::IndirectObjectDictionary (
    unsigned long int objNum = 0L,
    unsigned short int genNum = 0,
    CrossReferenceTable * tab = NULL ) [inline]
```

Constructor.

Create a new dictionary.

##### Parameters

<i>objNum</i>	The next object number.
<i>genNum</i>	The generation number.
<i>tab</i>	The cross reference table we are in.

#### 9.70.2.2 ~IndirectObjectDictionary()

```
FCFSupport::PDFFileStructures::IndirectObjectDictionary::~~IndirectObjectDictionary ( ) [inline]
```

Destructor.

Clean everything up.

### 9.70.3 Member Function Documentation

#### 9.70.3.1 AddIndirectObject()

```
void FCFSupport::PDFFileStructures::IndirectObjectDictionary::AddIndirectObject (
    const string name,
    IndirectObject * obj ) [inline]
```

Add an indirect object.

## Parameters

<i>name</i>	The name of the object.
<i>obj</i>	The object itself.

## 9.70.3.2 Size()

```
int FCFSupport::PDFFileStructures::IndirectObjectDictionary::Size ( ) const [inline]
```

Return the number of elements in this dictionary.

## 9.70.3.3 WriteDictionaryElements()

```
virtual ostream& FCFSupport::PDFFileStructures::IndirectObjectDictionary::WriteDictionaryElements  
(  
    ostream & stream ) const [inline], [protected], [virtual]
```

Write the elements of a dictionary.

## Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

Reimplemented from [FCFSupport::PDFFileStructures::Dictionary](#).

References [FCFSupport::PDFFileStructures::IndirectObject::HasOffset\(\)](#), [lcc::stream](#), [FCFSupport::PDFFileStructures::IndirectObject::WriteDirect\(\)](#), and [FCFSupport::PDFFileStructures::IndirectObject::WriteIndirectReference\(\)](#).

## 9.70.4 Member Data Documentation

## 9.70.4.1 elements

```
NamedIndirectObjectMap FCFSupport::PDFFileStructures::IndirectObjectDictionary::elements [private]
```

The elements in this dictionary.

## 9.71 FCFSupport::Industry Class Reference

The [Industry](#) class represents an industry.

```
#include <Industry.h>
```

### Public Member Functions

- [Industry](#) ()  
*Default constructor.*
- [Industry](#) ([Industry](#) &other)  
*Copy constructor.*
- [Industry](#) & operator= ([Industry](#) &other)  
*Assignment operator.*
- [Industry](#) (char t, [Station](#) \*st, const char \*n, int tl, int al, int p, bool r, char h, [Industry](#) \*m, int pl, int c, const char \*dcl, int mcl, const char \*lt, const char \*et)  
*Full constructor.*
- char [Type](#) () const  
*Return the type of the industry.*
- [Station](#) \* [MyStation](#) () const  
*Return the industry's station.*
- const char \* [Name](#) () const  
*Return the industry's name.*
- int [TrackLen](#) () const  
*Return the amount of track at this industry.*
- int [AssignLen](#) () const  
*Return the assignable amount of track at this industry.*
- int [Priority](#) () const  
*Return this industry's priority.*
- bool [Reload](#) () const  
*Can this industry reload cars?*
- char [Hazard](#) () const  
*What sorts of hazardous material classes can this industry handle?*
- [Industry](#) \* [MyMirror](#) () const  
*This industry's mirror industry (if any).*
- int [MaxPlate](#) () const  
*Maximum clearance plate this industry can handle.*
- int [MaxWeightClass](#) () const  
*Maximum weight class this industry can handle.*
- const char \* [DivisionControlList](#) () const  
*This industry's division control list.*
- int [MaxCarLen](#) () const  
*The maximum car length this industry can handle.*
- const char \* [LoadsAccepted](#) () const  
*The types of loads this industry can handle.*
- const char \* [EmptiesAccepted](#) () const

*The types of empties this industry can handle.*

- [FCFSupport::Car \\* TheCar](#) (int i) const

*Return the indexed car at this industry.*

- int [NumberOfCars](#) () const

*Return the number of cars at this industry.*

- void [IncrementStatsLen](#) (int i=1)

*Increment the stats length.*

- int [CarsNum](#) () const

*Return the number of cars.*

- int [CarsLen](#) () const

*Return the length of all of the cars.*

- int [StatsLen](#) () const

*Return the stats length.*

## Private Attributes

- [CarVector cars](#)

*The vector of cars at this industry.*

- [Station \\* station](#)

*The station this industry is at.*

- [Industry \\* mirror](#)

*The mirror industry or NULL if there is no mirror industry.*

- string [name](#)

*The name of the industry.*

- string [loadTypes](#)

*The vector of loaded car type charactes.*

- string [emptyTypes](#)

*The vector of empty car type characters.*

- string [divisionControllist](#)

*The division control list.*

- int [trackLen](#)

*The track length.*

- int [assignLen](#)

*The assignable length.*

- int [priority](#)

*The industry's priority.*

- int [plate](#)

*The industry's clearance plate.*

- int [weightclass](#)

*The industry's weight class.*

- int [maxCarLen](#)

*The maximum car length.*

- int [carsNum](#)

*The number of cars.*

- int [carsLen](#)

*The length of the cars.*

- int [statsLen](#)  
*The stats length.*
- int [usedLen](#)  
*The used length.*
- int [remLen](#)  
*The remaining length.*
- bool [reload](#)  
*The reload flag.*
- char [type](#)  
*The industry type.*
- char [hazard](#)  
*The hazard type character.*

## Friends

- class [System](#)  
*The [System](#) class is a friend.*

### 9.71.1 Detailed Description

The [Industry](#) class represents an industry.

There are several types of industries, including yards, on line industries, and off line industries. An industry has track where cars can be spotted for storage, loading, and unloading. On-line industries and yards have this trackage on the layout. off line industries have this trackage either in the form of a hidden staging yard or don't have any real trackage at all.

An industry takes specific loaded and empty car types, has a maximum weight and clearance plate, in at a specific station and has a division control list. Some industries are mirrors of others and some industries can re-load cars.

```
@author Robert Heller \<heller\@deepsoft.com\>
```

### 9.71.2 Constructor & Destructor Documentation

#### 9.71.2.1 [Industry\(\)](#) [1/3]

```
FCFSupport::Industry::Industry ( ) [inline]
```

Default constructor.

Fill all slots with default values.

References [assignLen](#), [carsLen](#), [carsNum](#), [divisionControlList](#), [emptyTypes](#), [hazard](#), [loadTypes](#), [maxCarLen](#), [mirror](#), [name](#), [plate](#), [priority](#), [reload](#), [remLen](#), [station](#), [statsLen](#), [trackLen](#), [type](#), [usedLen](#), and [weightclass](#).



### 9.71.2.2 Industry() [2/3]

```
FCFSupport::Industry::Industry (  
    Industry & other ) [inline]
```

Copy constructor.

Initialize this industry from another existing industry.

## Parameters

<i>other</i>	The other industry.
--------------	---------------------

References assignLen, cars, carsLen, carsNum, divisionControlList, emptyTypes, hazard, loadTypes, maxCarLen, mirror, name, plate, priority, reload, remLen, station, statsLen, trackLen, type, usedLen, and weightclass.

## 9.71.2.3 Industry() [3/3]

```
FCFSupport::Industry::Industry (
    char t,
    Station * st,
    const char * n,
    int tl,
    int al,
    int p,
    bool r,
    char h,
    Industry * m,
    int pl,
    int c,
    const char * dcl,
    int mcl,
    const char * lt,
    const char * et ) [inline]
```

Full constructor.

Create a new industry from a full set of parameters.

## Parameters

<i>t</i>	The type of industry ('Y' for yard, 'O' for offline, 'I' for online).
<i>st</i>	<a href="#">Station</a> this industry is at.
<i>n</i>	The name of the industry.
<i>tl</i>	The track length at this industry.
<i>al</i>	The assignable length at this industry.
<i>p</i>	This industry's priority.
<i>r</i>	<a href="#">Car</a> reload flag.
<i>h</i>	Hazard code.
<i>m</i>	Mirror industry.
<i>pl</i>	Maximum clearance plate.
<i>c</i>	Maximum weight class.
<i>dcl</i>	<a href="#">Division</a> control list.
<i>mcl</i>	Maximum car length.
<i>lt</i>	Loaded car types accepted here.
<i>et</i>	Empty car type accepted here.

References assignLen, carsLen, carsNum, divisionControlList, emptyTypes, hazard, loadTypes, maxCarLen, mirror, name, plate, priority, reload, remLen, station, statsLen, trackLen, type, usedLen, and weightclass.

### 9.71.3 Member Function Documentation

#### 9.71.3.1 AssignLen()

```
int FCFSupport::Industry::AssignLen ( ) const [inline]
```

Return the assignable amount of track at this industry.

References assignLen.

#### 9.71.3.2 CarsLen()

```
int FCFSupport::Industry::CarsLen ( ) const [inline]
```

Return the length of all of the cars.

References carsLen.

#### 9.71.3.3 CarsNum()

```
int FCFSupport::Industry::CarsNum ( ) const [inline]
```

Return the number of cars.

References carsNum.

#### 9.71.3.4 DivisionControlList()

```
const char* FCFSupport::Industry::DivisionControlList ( ) const [inline]
```

This industry's division control list.

References divisionControlList.

#### 9.71.3.5 EmptiesAccepted()

```
const char* FCFSupport::Industry::EmptiesAccepted ( ) const [inline]
```

The types of empties this industry can handle.

References emptyTypes.

#### 9.71.3.6 Hazard()

```
char FCFSupport::Industry::Hazard ( ) const [inline]
```

What sorts of hazardous material classes can this industry handle?

References hazard.

#### 9.71.3.7 IncrementStatsLen()

```
void FCFSupport::Industry::IncrementStatsLen (
    int i = 1 ) [inline]
```

Increment the stats length.

References i, and statsLen.

#### 9.71.3.8 LoadsAccepted()

```
const char* FCFSupport::Industry::LoadsAccepted ( ) const [inline]
```

The types of loads this industry can handle.

References loadTypes.

#### 9.71.3.9 MaxCarLen()

```
int FCFSupport::Industry::MaxCarLen ( ) const [inline]
```

The maximum car length this industry can handle.

References maxCarLen.

#### 9.71.3.10 MaxPlate()

```
int FCFSupport::Industry::MaxPlate ( ) const [inline]
```

Maximum clearance plate this industry can handle.

References plate.

#### 9.71.3.11 MaxWeightClass()

```
int FCFSupport::Industry::MaxWeightClass ( ) const [inline]
```

Maximum weight class this industry can handle.

References weightclass.

#### 9.71.3.12 MyMirror()

```
Industry* FCFSupport::Industry::MyMirror ( ) const [inline]
```

This industry's mirror industry (if any).

References mirror.

#### 9.71.3.13 MyStation()

```
Station* FCFSupport::Industry::MyStation ( ) const [inline]
```

Return the industry's station.

References station.

Referenced by FCFSupport::SwitchListElement::DropStopStation().

#### 9.71.3.14 Name()

```
const char* FCFSupport::Industry::Name ( ) const [inline]
```

Return the industry's name.

References name.

#### 9.71.3.15 NumberOfCars()

```
int FCFSupport::Industry::NumberOfCars ( ) const [inline]
```

Return the number of cars at this industry.

References cars.

#### 9.71.3.16 operator=()

```
Industry& FCFSupport::Industry::operator= (
    Industry & other ) [inline]
```

Assignment operator.

Initialize this industry from another existing industry.

##### Parameters

<i>other</i>	The other industry.
--------------	---------------------

References assignLen, cars, carsLen, carsNum, divisionControlList, emptyTypes, hazard, loadTypes, maxCarLen, mirror, name, plate, priority, reload, remLen, station, statsLen, trackLen, type, usedLen, and weightclass.

#### 9.71.3.17 Priority()

```
int FCFSupport::Industry::Priority ( ) const [inline]
```

Return this industry's priority.

References priority.

#### 9.71.3.18 Reload()

```
bool FCFSupport::Industry::Reload ( ) const [inline]
```

Can this industry reload cars?

References reload.

#### 9.71.3.19 StatsLen()

```
int FCFSupport::Industry::StatsLen ( ) const [inline]
```

Return the stats length.

References statsLen.

#### 9.71.3.20 TheCar()

```
FCFSupport::Car* FCFSupport::Industry::TheCar (
    int i ) const [inline]
```

Return the indexed car at this industry.

##### Parameters

<i>i</i>	This car index.
----------	-----------------

References cars, and i.

#### 9.71.3.21 TrackLen()

```
int FCFSupport::Industry::TrackLen ( ) const [inline]
```

Return the amount of track at this industry.

References trackLen.

#### 9.71.3.22 Type()

```
char FCFSupport::Industry::Type ( ) const [inline]
```

Return the type of the industry.

References type.

### 9.71.4 Friends And Related Function Documentation

#### 9.71.4.1 System

```
friend class System [friend]
```

The [System](#) class is a friend.

### 9.71.5 Member Data Documentation

#### 9.71.5.1 assignLen

```
int FCFSupport::Industry::assignLen [private]
```

The assignable length.

Referenced by [AssignLen\(\)](#), [Industry\(\)](#), and [operator=\(\)](#).

#### 9.71.5.2 cars

```
CarVector FCFSupport::Industry::cars [private]
```

The vector of cars at this industry.

Referenced by [Industry\(\)](#), [NumberOfCars\(\)](#), [operator=\(\)](#), and [TheCar\(\)](#).

#### 9.71.5.3 carsLen

```
int FCFSupport::Industry::carsLen [private]
```

The length of the cars.

Referenced by [CarsLen\(\)](#), [Industry\(\)](#), and [operator=\(\)](#).

#### 9.71.5.4 carsNum

```
int FCFSupport::Industry::carsNum [private]
```

The number of cars.

Referenced by [CarsNum\(\)](#), [Industry\(\)](#), and [operator=\(\)](#).



#### 9.71.5.5 divisionControlList

```
string FCFSupport::Industry::divisionControlList [private]
```

The division control list.

Referenced by DivisionControlList(), Industry(), and operator=().

#### 9.71.5.6 emptyTypes

```
string FCFSupport::Industry::emptyTypes [private]
```

The vector of empty car type characters.

Referenced by EmptiesAccepted(), Industry(), and operator=().

#### 9.71.5.7 hazard

```
char FCFSupport::Industry::hazard [private]
```

The hazard type character.

Referenced by Hazard(), Industry(), and operator=().

#### 9.71.5.8 loadTypes

```
string FCFSupport::Industry::loadTypes [private]
```

The vector of loaded car type charactes.

Referenced by Industry(), LoadsAccepted(), and operator=().

#### 9.71.5.9 maxCarLen

```
int FCFSupport::Industry::maxCarLen [private]
```

The maximum car length.

Referenced by Industry(), MaxCarLen(), and operator=().

#### 9.71.5.10 mirror

```
Industry* FCFSupport::Industry::mirror [private]
```

The mirror industry or NULL if there is no mirror industry.

Referenced by `Industry()`, `MyMirror()`, and `operator=()`.

#### 9.71.5.11 name

```
string FCFSupport::Industry::name [private]
```

The name of the industry.

Referenced by `Industry()`, `Name()`, and `operator=()`.

#### 9.71.5.12 plate

```
int FCFSupport::Industry::plate [private]
```

The industry's clearance plate.

Referenced by `Industry()`, `MaxPlate()`, and `operator=()`.

#### 9.71.5.13 priority

```
int FCFSupport::Industry::priority [private]
```

The industry's priority.

Referenced by `Industry()`, `operator=()`, and `Priority()`.

#### 9.71.5.14 reload

```
bool FCFSupport::Industry::reload [private]
```

The reload flag.

Referenced by `Industry()`, `operator=()`, and `Reload()`.

#### 9.71.5.15 remLen

```
int FCFSupport::Industry::remLen [private]
```

The remaining length.

Referenced by Industry(), and operator=().

#### 9.71.5.16 station

```
Station* FCFSupport::Industry::station [private]
```

The station this industry is at.

Referenced by Industry(), MyStation(), and operator=().

#### 9.71.5.17 statsLen

```
int FCFSupport::Industry::statsLen [private]
```

The stats length.

Referenced by IncrementStatsLen(), Industry(), operator=(), and StatsLen().

#### 9.71.5.18 trackLen

```
int FCFSupport::Industry::trackLen [private]
```

The track length.

Referenced by Industry(), operator=(), and TrackLen().

#### 9.71.5.19 type

```
char FCFSupport::Industry::type [private]
```

The industry type.

Referenced by Industry(), operator=(), and Type().

### 9.71.5.20 usedLen

```
int FCFSupport::Industry::usedLen [private]
```

The used length.

Referenced by Industry(), and operator=().

### 9.71.5.21 weightclass

```
int FCFSupport::Industry::weightclass [private]
```

The industry's weight class.

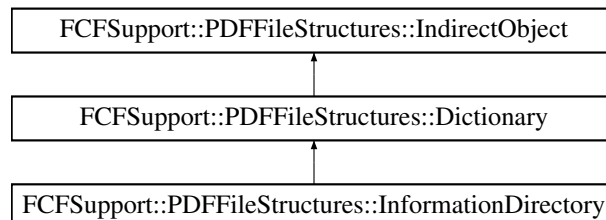
Referenced by Industry(), MaxWeightClass(), and operator=().

## 9.72 FCFSupport::PDFFileStructures::InformationDirectory Class Reference

Information directory.

```
#include <PDFPrinterSupport.h>
```

Inheritance diagram for FCFSupport::PDFFileStructures::InformationDirectory:



### Public Member Functions

- [InformationDirectory](#) (unsigned long int objNum=0L, unsigned short int genNum=0, CrossReferenceTable \*tab=↔ NULL)  
*Constructor.*
- [~InformationDirectory](#) ()  
*Destructor.*

## Public Attributes

- string [title](#)  
*The title.*
- string [author](#)  
*The author.*
- string [subject](#)  
*The subject.*
- string [keywords](#)  
*The keywords.*
- string [creator](#)  
*The creator.*
- string [producer](#)  
*The producer.*
- time\_t [creationDate](#)  
*The creationDate.*
- time\_t [modificationDate](#)  
*The modificationDate.*

## Protected Member Functions

- virtual ostream & [WriteDictionaryElements](#) (ostream &stream) const  
*Write an object directly.*

### 9.72.1 Detailed Description

Information directory.

Contains random extra information about the document.

Author

Robert Heller <heller@deepsoft.com>

### 9.72.2 Constructor & Destructor Documentation

#### 9.72.2.1 InformationDirectory()

```
FCFSupport::PDFFileStructures::InformationDirectory::InformationDirectory (  
    unsigned long int objNum = 0L,  
    unsigned short int genNum = 0,  
    CrossReferenceTable * tab = NULL ) [inline]
```

Constructor.

Create a basic information directory.

## Parameters

<i>objNum</i>	The next object number.
<i>genNum</i>	The generation number.
<i>tab</i>	The cross reference table we are in.

## 9.72.2.2 ~InformationDirectory()

```
FCFSupport::PDFFileStructures::InformationDirectory::~~InformationDirectory ( ) [inline]
```

Destructor.

## 9.72.3 Member Function Documentation

## 9.72.3.1 WriteDictionaryElements()

```
virtual ostream& FCFSupport::PDFFileStructures::InformationDirectory::WriteDictionaryElements (
    ostream & stream ) const [protected], [virtual]
```

Write an object directly.

## Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

Reimplemented from [FCFSupport::PDFFileStructures::Dictionary](#).

## 9.72.4 Member Data Documentation

## 9.72.4.1 author

```
string FCFSupport::PDFFileStructures::InformationDirectory::author
```

The author.

#### 9.72.4.2 creator

```
string FCFSupport::PDFFileStructures::InformationDirectory::creator
```

The creator.

#### 9.72.4.3 creationDate

```
time_t FCFSupport::PDFFileStructures::InformationDirectory::creationDate
```

The creationDate.

#### 9.72.4.4 keywords

```
string FCFSupport::PDFFileStructures::InformationDirectory::keywords
```

The keywords.

#### 9.72.4.5 modificationDate

```
time_t FCFSupport::PDFFileStructures::InformationDirectory::modificationDate
```

The modificationDate.

#### 9.72.4.6 producer

```
string FCFSupport::PDFFileStructures::InformationDirectory::producer
```

The producer.

#### 9.72.4.7 subject

```
string FCFSupport::PDFFileStructures::InformationDirectory::subject
```

The subject.

## 9.72.4.8 title

```
string FCFSupport::PDFFileStructures::InformationDirectory::title
```

The title.

## 9.73 Parsers::IntegerList Class Reference

The [IntegerList](#) class implements a linked list of integers, used for turnout route lists.

```
#include <IntegerList.h>
```

### Public Member Functions

- [IntegerList](#) (int car=0, [IntegerList](#) \*cdr=NULL)  
*Base constructor.*
- int [Element](#) () const  
*Element accessor.*
- const [IntegerList](#) \* [Next](#) () const  
*Next pointer accessor (Const version).*
- [IntegerList](#) \* [Next](#) ()  
*Next pointer accessor (non-Const version).*
- bool [ElementP](#) (int v) const  
*Is value in the list?*

### Static Public Member Functions

- static [IntegerList](#) \* [IntAppend](#) ([IntegerList](#) \*head, int newTail)  
*Add an element to the {end} of the list.*
- static void [CleanUpIntegerList](#) ([IntegerList](#) \*list)  
*Free up used memory.*
- static [IntegerList](#) \* [CopyList](#) (const [IntegerList](#) \*src)

### Private Attributes

- int [iElt](#)  
*The current element.*
- [IntegerList](#) \* [next](#)  
*The pointer to the next element.*

### Friends

- class [TurnoutBodyElt](#)
- ostream & [operator<<](#) (ostream &stream, [IntegerList](#) list)  
*Output operator.*



### 9.73.1 Detailed Description

The `IntegerList` class implements a linked list of integers, used for turnout route lists.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.73.2 Constructor & Destructor Documentation

#### 9.73.2.1 IntegerList()

```
Parsers::IntegerList::IntegerList (
    int car = 0,
    IntegerList * cdr = NULL ) [inline]
```

Base constructor.

Referenced by `IntAppend()`.

### 9.73.3 Member Function Documentation

#### 9.73.3.1 CleanUpIntegerList()

```
static void Parsers::IntegerList::CleanUpIntegerList (
    IntegerList * list ) [inline], [static]
```

Free up used memory.

References next.

#### 9.73.3.2 CopyList()

```
static IntegerList* Parsers::IntegerList::CopyList (
    const IntegerList * src ) [inline], [static]
```

References `iElt`, `IntAppend()`, and `next`.

### 9.73.3.3 Element()

```
int Parsers::IntegerList::Element ( ) const [inline]
```

Element accessor.

References iElt.

### 9.73.3.4 ElementP()

```
bool Parsers::IntegerList::ElementP (
    int v ) const [inline]
```

Is value in the list?

References iElt, and next.

### 9.73.3.5 IntAppend()

```
static IntegerList* Parsers::IntegerList::IntAppend (
    IntegerList * head,
    int newTail ) [inline], [static]
```

Add an element to the *{end}* of the list.

References IntegerList(), and next.

Referenced by CopyList().

### 9.73.3.6 Next() [1/2]

```
const IntegerList* Parsers::IntegerList::Next ( ) const [inline]
```

Next pointer accessor (Const version).

References next.

### 9.73.3.7 Next() [2/2]

```
IntegerList* Parsers::IntegerList::Next ( ) [inline]
```

Next pointer accessor (non-Const version).

References next.

## 9.73.4 Friends And Related Function Documentation

### 9.73.4.1 operator<<

```
ostream& operator<< (
    ostream & stream,
    IntegerList list ) [friend]
```

Output operator.

### 9.73.4.2 TurnoutBodyElt

```
friend class TurnoutBodyElt [friend]
```

## 9.73.5 Member Data Documentation

### 9.73.5.1 iElt

```
int Parsers::IntegerList::iElt [private]
```

The current element.

Referenced by CopyList(), Element(), and ElementP().

## 9.73.5.2 next

```
IntegerList* Parsers::IntegerList::next [private]
```

The pointer to the next element.

Referenced by CleanUpIntegerList(), CopyList(), ElementP(), IntAppend(), and Next().

## 9.74 CTCPanel::Lamp Class Reference

Lamp object type.

### Public Member Functions

- [Lamp](#) (name, \_ctcpanel, \_canvas,...)  
*Construct a [Lamp](#) object.*
- [~Lamp](#) ()  
*Clean up all data objects and free up all resources.*
- [getv](#) ()  
*Method to get our value (lamp state).*
- [setv](#) (newstate)  
*Method to set out value (lamp state).*
- [geti](#) (ind)  
*Method to get the state of one of our indicators (none).*
- [seti](#) (ind, value)  
*Method to set an indicator's state (none).*
- [invoke](#) ()  
*Method to invoke the lamp.*

### Private Member Functions

- [\\_configureColor](#) (option, value)  
*Method to update the lamp color.*
- [\\_configureLabel](#) (option, value)  
*Method to update the label option.*

### Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*
- [state](#)  
*The state of the lamp.*

#### 9.74.1 Detailed Description

Lamp object type.

These are on the control panel and represent simple single-color lamps.

## Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The control panel canvas to draw the lamp on.
<code>...</code>	Options: <ul style="list-style-type: none"> <li>• <code>-x</code> The x coordinate of the object (readonly, default 0).</li> <li>• <code>-y</code> The y coordinate of the object (readonly, default 0).</li> <li>• <code>-controlpoint</code> The name of the control point this lamp is part of (readonly, default CP1).</li> <li>• <code>-color</code> The color of the lamp (default white).</li> <li>• <code>-label</code> The label of the lamp (default "lamp").</li> </ul>

Defined coords terminals: none. Defined values (states):

- on [Lamp](#) is on.
- off [Lamp](#) is off.

Defined indicators: none.

## Author

Robert Heller <heller@deepsoft.com>

## 9.74.2 Constructor &amp; Destructor Documentation

## 9.74.2.1 Lamp()

```
CTCPanel::Lamp::Lamp (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a [Lamp](#) object.

## Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The control panel canvas to draw the <a href="#">Lamp</a> on.
<code>...</code>	Option list.

### 9.74.2.2 ~Lamp()

```
CTCPanel::Lamp::~~Lamp ( )
```

Clean up all data objects and free up all resources.

## 9.74.3 Member Function Documentation

### 9.74.3.1 \_configureColor()

```
CTCPanel::Lamp::_configureColor (
    option ,
    value ) [private]
```

Method to update the lamp color.

### 9.74.3.2 \_configureLabel()

```
CTCPanel::Lamp::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

### 9.74.3.3 geti()

```
CTCPanel::Lamp::geti (
    ind )
```

Method to get the state of one of our indicators (none).

#### 9.74.3.4 getv()

```
CTCPanel::Lamp::getv ( )
```

Method to get our value (lamp state).

#### 9.74.3.5 invoke()

```
CTCPanel::Lamp::invoke ( )
```

Method to invoke the lamp.

#### 9.74.3.6 seti()

```
CTCPanel::Lamp::seti (
    ind ,
    value )
```

Method to set an indicator's state (none).

#### 9.74.3.7 setv()

```
CTCPanel::Lamp::setv (
    newstate )
```

Method to set out value (lamp state).

##### Parameters

<i>newstate</i>	The new lamp state.
-----------------	---------------------

### 9.74.4 Member Data Documentation

#### 9.74.4.1 canvas

```
CTCPanel::Lamp::canvas [private]
```

The canvas component (parent widget component).

#### 9.74.4.2 ctcpanel

```
CTCPanel::Lamp::ctcpanel [private]
```

The CTC Panel component (parent widget).

#### 9.74.4.3 state

```
CTCPanel::Lamp::state [private]
```

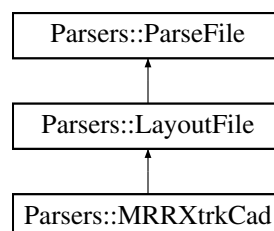
The state of the lamp.

## 9.75 Parsers::LayoutFile Class Reference

File to parse an XTrkCad layout file and create a track graph.

```
#include <ParseFile.h>
```

Inheritance diagram for Parsers::LayoutFile:





## Public Member Functions

- [LayoutFile](#) (const char \*filename, [MRRXtrkCad](#) \*p)  
*Constructor.*
- virtual [~LayoutFile](#) ()  
*Destructor.*
- void [Emit](#) (ostream &ostream)  
*Function to Emit a track graph to an output stream.*
- bool [IsNodeP](#) (int nid) const  
*Tests if a node id exists in the graph.*
- int [NumEdges](#) (int nid) const  
*Returns the number of edges for the specified node id.*
- int [EdgeIndex](#) (int nid, int edgenum) const  
*Returns the node id of the specified edge of the node.*
- float [EdgeX](#) (int nid, int edgenum) const  
*Returns the \$X\$ coordinate of the specified edge of the node.*
- float [EdgeY](#) (int nid, int edgenum) const  
*Returns the \$Y\$ coordinate of the specified edge of the node.*
- float [EdgeA](#) (int nid, int edgenum) const  
*Returns the angle of the specified edge of the node.*
- float [EdgeLength](#) (int nid, int edgenum) const  
*Returns the length of an edge.*
- [TrackGraph::NodeType](#) [TypeOfNode](#) (int nid) const  
*Returns the type of the node.*
- const [TurnoutGraphic](#) \* [NodeTurnoutGraphic](#) (int nid) const  
*Returns the [TurnoutGraphic](#) of the node.*
- const [TurnoutRoutelist](#) \* [NodeTurnoutRoutelist](#) (int nid) const  
*Returns the [TurnoutRoutelist](#) of the node.*
- float [LengthOfNode](#) (int nid) const  
*Return the track length of a node.*
- const [IntegerList](#) \* [TrackList](#) (int nid) const  
*Return a block's tracklist.*
- int [TurnoutNumber](#) (int nid) const  
*Return a switchmotor's turnout number.*
- const char \* [NameOfNode](#) (int nid) const  
*Return a block's or switchmotor's name.*
- const char \* [SenseScript](#) (int nid) const  
*Return a block's or switchmotor's sense script.*
- const char \* [NormalActionScript](#) (int nid) const  
*Return a switchmotor's normal action script.*
- const char \* [ReverseActionScript](#) (int nid) const  
*Return a block's or switchmotor's reverse action script.*
- int [NumberOfHeads](#) (int nid) const  
*Return a Signal's number of heads.*
- const [StringPairList](#) \* [SignalAspects](#) (int nid) const  
*Return a Signal's aspect list.*
- float [OrigX](#) (int nid) const

- Return the  $x$  coordinate of the signal base.*

  - float [OrigY](#) (int nid) const
- Return the  $y$  coordinate of the signal base.*

  - float [Angle](#) (int nid) const
- Return the angle of the signal.*

  - const char \* [OnScript](#) (int nid) const
- Return the on action script.*

  - const char \* [OffScript](#) (int nid) const
- Return the off action script.*

  - int [LowestNode](#) () const
- Returns the lowest numbered node id.*

  - int [HighestNode](#) () const
- Returns the highest numbered node id.*

  - void [CompressGraph](#) ()
- Create a compressed graph.*

  - int [CompressedEdgeCount](#) (int cnid) const
- Number of compressed graph edges.*

  - float [CompressedEdgeLength](#) (int cnid, int edgenum) const
- Length of a compressed graph edge.*

  - int [CompressedEdgeNode](#) (int cnid, int edgenum) const
- Next Edge node.*

  - [IntegerList](#) \* [CompressedNodeSegments](#) (int cnid) const
- Raw nodes in a compressed graph node.*

  - bool [IsCompressed](#) () const
- const [IntegerList](#) \* [Heads](#) ()
- bool [IsCompressedNode](#) (int cnid) const
- Is cid a node in the compressed graph?*

  - const [IntegerList](#) \* [Roots](#) ()
- Compressed graph roots.*

  - double [CompressedNodePositionX](#) (int cnid) const
- X Coordinate of a Compressed Node position.*

  - double [CompressedNodePositionY](#) (int cnid) const
- X Coordinate of a Compressed Node position.*

  - void [CompressedGraphCircleLayout](#) (double radius)
- Run the BGL circle\_graph\_layout for a given radius.*

  - bool [CompressedGraphKamadaKawaiSpring](#) (double sidelength)
- Run the BGL kamada\_kawai\_spring\_layout for a given side length.*

  - [TrackGraph::CompressedEdgePairVector](#) [CompressedGraphKruskalMinimumSpanningTree](#) ()
- Run the kruskal\_minimum\_spanning\_tree algorithm and return a vector of edge pairs.*

  - [TrackGraph::CompressedEdgePairVector](#) [CompressedGraphPrimMinimumSpanningTree](#) ()
- Run the prim\_minimum\_spanning\_tree algorithm and return a vector of edge pairs.*

## Protected Member Functions

- virtual int [Parse](#) ()

*Parseer function.*
- virtual void [ParseError](#) (const char \*m)

*Parse error handler.*

## Protected Attributes

- [MRRXtrkCad](#) \* [parser](#)  
*Parser.*
- [TrackGraph](#) \* [trackGraph](#)  
*Track graph, a graph of all of the trackwork in the layout file.*

## Additional Inherited Members

### 9.75.1 Detailed Description

File to parse an XTrkCad layout file and create a track graph.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.75.2 Constructor & Destructor Documentation

#### 9.75.2.1 LayoutFile()

```
Parsers::LayoutFile::LayoutFile (
    const char * filename,
    MRRXtrkCad * p )
```

Constructor.

See [MRRXtrkCad](#). User code normally creates a [MRRXtrkCad](#) object and the [MRRXtrkCad](#) constructor calls this constructor. User code then accesses the inherited methods of [LayoutFile](#) and [ParseFile](#) from the [MRRXtrkCad](#) object.

#### 9.75.2.2 ~LayoutFile()

```
virtual Parsers::LayoutFile::~~LayoutFile ( ) [virtual]
```

Destructor.

### 9.75.3 Member Function Documentation

#### 9.75.3.1 Angle()

```
float Parsers::LayoutFile::Angle (
    int nid ) const [inline]
```

Return the angle of the signal.

#### Parameters

<i>nid</i>	The node to look at.
------------	----------------------

References `Parsers::TrackGraph::Angle()`.

#### 9.75.3.2 `CompressedEdgeCount()`

```
int Parsers::LayoutFile::CompressedEdgeCount (  
    int cnid ) const [inline]
```

Number of compressed graph edges.

References `Parsers::TrackGraph::CompressedEdgeCount()`.

#### 9.75.3.3 `CompressedEdgeLength()`

```
float Parsers::LayoutFile::CompressedEdgeLength (  
    int cnid,  
    int edgenum ) const [inline]
```

Length of a compressed graph edge.

References `Parsers::TrackGraph::CompressedEdgeLength()`.

#### 9.75.3.4 `CompressedEdgeNode()`

```
int Parsers::LayoutFile::CompressedEdgeNode (  
    int cnid,  
    int edgenum ) const [inline]
```

Next Edge node.

References `Parsers::TrackGraph::CompressedEdgeNode()`.

#### 9.75.3.5 CompressedGraphCircleLayout()

```
void Parsers::LayoutFile::CompressedGraphCircleLayout (
    double radius ) [inline]
```

Run the BGL circle\_graph\_layout for a given radius.

References Parsers::TrackGraph::CompressedGraphCircleLayout().

#### 9.75.3.6 CompressedGraphKamadaKawaiSpring()

```
bool Parsers::LayoutFile::CompressedGraphKamadaKawaiSpring (
    double sidelength ) [inline]
```

Run the BGL kamada\_kawai\_spring\_layout for a given side length.

References Parsers::TrackGraph::CompressedGraphKamadaKawaiSpring().

#### 9.75.3.7 CompressedGraphKruskalMinimumSpanningTree()

```
TrackGraph::CompressedEdgePairVector Parsers::LayoutFile::CompressedGraphKruskalMinimumSpanningTree (
    ) [inline]
```

Run the kruskal\_minimum\_spanning\_tree algorithm and return a vector of edge pairs.

References Parsers::TrackGraph::CompressedGraphKruskalMinimumSpanningTree().

#### 9.75.3.8 CompressedGraphPrimMinimumSpanningTree()

```
TrackGraph::CompressedEdgePairVector Parsers::LayoutFile::CompressedGraphPrimMinimumSpanningTree (
    ) [inline]
```

Run the prim\_minimum\_spanning\_tree algorithm and return a vector of edge pairs.

References Parsers::TrackGraph::CompressedGraphPrimMinimumSpanningTree().

#### 9.75.3.9 CompressedNodePositionX()

```
double Parsers::LayoutFile::CompressedNodePositionX (
    int cnid ) const [inline]
```

X Coordinate of a Compressed Node position.

References Parsers::TrackGraph::CompressedNodePositionX().

#### 9.75.3.10 CompressedNodePositionY()

```
double Parsers::LayoutFile::CompressedNodePositionY (
    int cnid ) const [inline]
```

X Coordinate of a Compressed Node position.

References Parsers::TrackGraph::CompressedNodePositionY().

#### 9.75.3.11 CompressedNodeSegments()

```
IntegerList* Parsers::LayoutFile::CompressedNodeSegments (
    int cnid ) const [inline]
```

Raw nodes in a compressed graph node.

References Parsers::TrackGraph::CompressedNodeSegments().

#### 9.75.3.12 CompressGraph()

```
void Parsers::LayoutFile::CompressGraph ( ) [inline]
```

Create a compressed graph.

References Parsers::TrackGraph::CompressGraph().

#### 9.75.3.13 EdgeA()

```
float Parsers::LayoutFile::EdgeA (
    int nid,
    int edgenum ) const [inline]
```

Returns the angle of the specified edge of the node.

References Parsers::TrackGraph::EdgeA().

#### 9.75.3.14 EdgeIndex()

```
int Parsers::LayoutFile::EdgeIndex (
    int nid,
    int edgenum ) const [inline]
```

Returns the node id of the specified edge of the node.

References Parsers::TrackGraph::EdgeIndex().

#### 9.75.3.15 EdgeLength()

```
float Parsers::LayoutFile::EdgeLength (
    int nid,
    int edgenum ) const [inline]
```

Returns the length of an edge.

References Parsers::TrackGraph::EdgeLength().

#### 9.75.3.16 EdgeX()

```
float Parsers::LayoutFile::EdgeX (
    int nid,
    int edgenum ) const [inline]
```

Returns the  $X$  coordinate of the specified edge of the node.

References Parsers::TrackGraph::EdgeX().

### 9.75.3.17 EdgeY()

```
float Parsers::LayoutFile::EdgeY (
    int nid,
    int edgenum ) const [inline]
```

Returns the  $Y$  coordinate of the specified edge of the node.

References Parsers::TrackGraph::EdgeY().

### 9.75.3.18 Emit()

```
void Parsers::LayoutFile::Emit (
    ostream & ostream )
```

Function to Emit a track graph to an output stream.

#### Parameters

<i>ostream</i>	The output stream to write the graph to.
----------------	--

### 9.75.3.19 Heads()

```
const IntegerList* Parsers::LayoutFile::Heads ( ) [inline]
```

References Parsers::TrackGraph::Heads().

### 9.75.3.20 HighestNode()

```
int Parsers::LayoutFile::HighestNode ( ) const [inline]
```

Returns the highest numbered node id.

References Parsers::TrackGraph::HighestNode().



#### 9.75.3.21 IsCompressed()

```
bool Parsers::LayoutFile::IsCompressed ( ) const [inline]
```

References Parsers::TrackGraph::IsCompressed().

#### 9.75.3.22 IsCompressedNode()

```
bool Parsers::LayoutFile::IsCompressedNode (
    int cnid ) const [inline]
```

Is *cid* a node in the compressed graph?

References Parsers::TrackGraph::IsCompressedNode().

#### 9.75.3.23 IsNodeP()

```
bool Parsers::LayoutFile::IsNodeP (
    int nid ) const [inline]
```

Tests if a node *id* exists in the graph.

References Parsers::TrackGraph::IsNodeP().

#### 9.75.3.24 LengthOfNode()

```
float Parsers::LayoutFile::LengthOfNode (
    int nid ) const [inline]
```

Return the track length of a node.

References Parsers::TrackGraph::LengthOfNode().

#### 9.75.3.25 LowestNode()

```
int Parsers::LayoutFile::LowestNode ( ) const [inline]
```

Returns the lowest numbered node *id*.

References Parsers::TrackGraph::LowestNode().

#### 9.75.3.26 NameOfNode()

```
const char* Parsers::LayoutFile::NameOfNode (
    int nid ) const [inline]
```

Return a block's or switchmotor's name.

References Parsers::TrackGraph::NameOfNode().

#### 9.75.3.27 NodeTurnoutGraphic()

```
const TurnoutGraphic* Parsers::LayoutFile::NodeTurnoutGraphic (
    int nid ) const [inline]
```

Returns the [TurnoutGraphic](#) of the node.

References Parsers::TrackGraph::NodeTurnoutGraphic().

#### 9.75.3.28 NodeTurnoutRoutelist()

```
const TurnoutRoutelist* Parsers::LayoutFile::NodeTurnoutRoutelist (
    int nid ) const [inline]
```

Returns the [TurnoutRoutelist](#) of the node.

References Parsers::TrackGraph::NodeTurnoutRoutelist().

#### 9.75.3.29 NormalActionScript()

```
const char* Parsers::LayoutFile::NormalActionScript (
    int nid ) const [inline]
```

Return a switchmotor's normal action script.

References Parsers::TrackGraph::NormalActionScript().

#### 9.75.3.30 NumberOfHeads()

```
int Parsers::LayoutFile::NumberOfHeads (
    int nid ) const [inline]
```

Return a Signal's number of heads.

References Parsers::TrackGraph::NumberOfHeads().

#### 9.75.3.31 NumEdges()

```
int Parsers::LayoutFile::NumEdges (
    int nid ) const [inline]
```

Returns the number of edges for the specified node id.

References Parsers::TrackGraph::NumEdges().

#### 9.75.3.32 OffScript()

```
const char* Parsers::LayoutFile::OffScript (
    int nid ) const [inline]
```

Return the off action script.

##### Parameters

<i>nid</i>	The node to look at.
------------	----------------------

References Parsers::TrackGraph::OffScript().

#### 9.75.3.33 OnScript()

```
const char* Parsers::LayoutFile::OnScript (
    int nid ) const [inline]
```

Return the on action script.

**Parameters**

<i>nid</i>	The node to look at.
------------	----------------------

References [Parsers::TrackGraph::OnScript\(\)](#).

**9.75.3.34 OrigX()**

```
float Parsers::LayoutFile::OrigX (  
    int nid ) const [inline]
```

Return the  $x$  coordinate of the signal base.

**Parameters**

<i>nid</i>	The node to look at.
------------	----------------------

References [Parsers::TrackGraph::OrigX\(\)](#).

**9.75.3.35 OrigY()**

```
float Parsers::LayoutFile::OrigY (  
    int nid ) const [inline]
```

Return the  $y$  coordinate of the signal base.

**Parameters**

<i>nid</i>	The node to look at.
------------	----------------------

References [Parsers::TrackGraph::OrigY\(\)](#).

**9.75.3.36 Parse()**

```
virtual int Parsers::LayoutFile::Parse ( ) [protected], [virtual]
```

Parser function.

Implements [Parsers::ParseFile](#).

#### 9.75.3.37 ParseError()

```
virtual void Parsers::LayoutFile::ParseError (
    const char * m ) [protected], [virtual]
```

Parse error handler.

Implements [Parsers::ParseFile](#).

#### 9.75.3.38 ReverseActionScript()

```
const char* Parsers::LayoutFile::ReverseActionScript (
    int nid ) const [inline]
```

Return a block's or switchmotor's reverse action script.

References [Parsers::TrackGraph::ReverseActionScript\(\)](#).

#### 9.75.3.39 Roots()

```
const IntegerList* Parsers::LayoutFile::Roots ( ) [inline]
```

Compressed graph roots.

References [Parsers::TrackGraph::Roots\(\)](#).

#### 9.75.3.40 SenseScript()

```
const char* Parsers::LayoutFile::SenseScript (
    int nid ) const [inline]
```

Return a block's or switchmotor's sense script.

References [Parsers::TrackGraph::SenseScript\(\)](#).

#### 9.75.3.41 SignalAspects()

```
const StringPairList* Parsers::LayoutFile::SignalAspects (
    int nid ) const [inline]
```

Return a Signal's aspect list.

References Parsers::TrackGraph::SignalAspects().

#### 9.75.3.42 TrackList()

```
const IntegerList* Parsers::LayoutFile::TrackList (
    int nid ) const [inline]
```

Return a block's tracklist.

References Parsers::TrackGraph::TrackList().

#### 9.75.3.43 TurnoutNumber()

```
int Parsers::LayoutFile::TurnoutNumber (
    int nid ) const [inline]
```

Return a switchmotor's turnout number.

References Parsers::TrackGraph::TurnoutNumber().

#### 9.75.3.44 TypeOfNode()

```
TrackGraph::NodeType Parsers::LayoutFile::TypeOfNode (
    int nid ) const [inline]
```

Returns the type of the node.

References Parsers::TrackGraph::TypeOfNode().

### 9.75.4 Member Data Documentation

#### 9.75.4.1 parser

`MRRXtrkCad* Parsers::LayoutFile::parser` [protected]

Parser.

#### 9.75.4.2 trackGraph

`TrackGraph* Parsers::LayoutFile::trackGraph` [protected]

Track graph, a graph of all of the trackwork in the layout file.

## 9.76 xpressnet::LI100Message Class Reference

LI100 messages.

### Public Member Functions

- `LI100Message` (name, mbyte)  
*Constructor.*
- `MessageType` ()  
*Return the message type.*

### Private Attributes

- `_message_type`  
*The message type.*

### 9.76.1 Detailed Description

LI100 messages.

Author

Robert Heller <heller@deepsoft.com>

### 9.76.2 Constructor & Destructor Documentation

#### 9.76.2.1 LI100Message()

```
xpressnet::LI100Message::LI100Message (  
    name ,  
    mbyte )
```

Constructor.

**Parameters**

<i>mbyte</i>	Message byte.
--------------	---------------

### 9.76.3 Member Function Documentation

#### 9.76.3.1 MessageType()

```
xpressnet::LI100Message::MessageType ( )
```

Return the message type.

### 9.76.4 Member Data Documentation

#### 9.76.4.1 \_message\_type

```
xpressnet::LI100Message::_message_type [private]
```

The message type.

## 9.77 xpressnet::LI100VersionNumbers Class Reference

LI100 Version Numbers.

### Public Member Functions

- [LI100VersionNumbers](#) (name, hv, sv)  
*Constructor.*
- [HardwareVersion](#) ()  
*Return hardware version.*
- [SoftwareVersion](#) ()  
*Return software version.*



## Private Attributes

- [\\_hardware\\_version](#)  
*Hardware version.*
- [\\_software\\_version](#)  
*Software version.*

### 9.77.1 Detailed Description

LI100 Version Numbers.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.77.2 Constructor & Destructor Documentation

#### 9.77.2.1 LI100VersionNumbers()

```
xpressnet::LI100VersionNumbers::LI100VersionNumbers (
    name ,
    hv ,
    sv )
```

Constructor.

#### Parameters

<i>mbyte</i>	Message byte.
--------------	---------------

### 9.77.3 Member Function Documentation

#### 9.77.3.1 HardwareVersion()

```
xpressnet::LI100VersionNumbers::HardwareVersion ( )
```

Return hardware version.

### 9.77.3.2 SoftwareVersion()

```
xpressnet::LI100VersionNumbers::SoftwareVersion ( )
```

Return software version.

## 9.77.4 Member Data Documentation

### 9.77.4.1 \_hardware\_version

```
xpressnet::LI100VersionNumbers::_hardware_version [private]
```

Hardware version.

### 9.77.4.2 \_software\_version

```
xpressnet::LI100VersionNumbers::_software_version [private]
```

Software version.

## 9.78 xpressnet::LI101XPressNetAddress Class Reference

LI101 XPress Net Address.

### Public Member Functions

- [LI101XPressNetAddress](#) (name, addr)  
*Constructor.*
- [Address](#) ()  
*Return [XPressNet](#) address.*

### Private Attributes

- [\\_address](#)  
*Address.*

### 9.78.1 Detailed Description

LI101 XPress Net Address.

Author

Robert Heller <heller@deepsoft.com>

### 9.78.2 Constructor & Destructor Documentation

#### 9.78.2.1 LI101XPressNetAddress()

```
xpressnet::LI101XPressNetAddress::LI101XPressNetAddress (
    name ,
    addr )
```

Constructor.

Parameters

<i>mbyte</i>	Message byte.
--------------	---------------

### 9.78.3 Member Function Documentation

#### 9.78.3.1 Address()

```
xpressnet::LI101XPressNetAddress::Address ( )
```

Return [XPressNet](#) address.

### 9.78.4 Member Data Documentation

#### 9.78.4.1 \_address

```
xpressnet::LI101XPressNetAddress::_address [private]
```

Address.

## 9.79 linuxgpio::LinuxGpio Class Reference

Base generic GPIO interface class.

### Public Member Functions

- [LinuxGpio](#) (name,...)  
*Constructor, used to set up the GPIO pin.*
- [read](#) ()  
*Read the value of the pin.*
- [write](#) (value)  
*Write value to the pin.*
- [is\\_output](#) ()  
*Returns a boolean value indicating whether the pin is an output pin or not.*
- [~LinuxGpio](#) ()  
*Destructor.*
- [Set](#) ()  
*Set the pin to logic true.*
- [Clr](#) ()  
*Set the pin to logic false.*
- [Get](#) ()  
*Get the pin's logic state.*

### Static Private Attributes

- static [EXPORT](#)  
*The name of the export control file.*
- static [UNEXPORT](#)  
*The name of the unexport control file.*
- static [DIRECTIONFMT](#)  
*The format string to generate the name of the direction control file.*
- static [VALUEFMT](#)  
*The format string to generate the name of the value file.*

### 9.79.1 Detailed Description

Base generic GPIO interface class.

(Use one of the specialized classes.)

This class implements the basic interface for a GPIO pin. The pin is set up, its direction configured and its value is optionally initialized.

## 9.79.2 Constructor & Destructor Documentation

### 9.79.2.1 LinuxGpio()

```
linuxgpio::LinuxGpio::LinuxGpio (
    name ,
    ... )
```

Constructor, used to set up the GPIO pin.

The pin number is written to the export control file and then the pin's direction control file is computed and the pin's direction is written.

#### Parameters

<i>name</i>	The name of the pin.
...	Options: <ul style="list-style-type: none"><li>• -pinnumber The pin number, readonly, defaults to 0 and can be any positive integer.</li><li>• -direction The pin direction, readonly, defaults to in can be one of in, out, high, or low.</li></ul>

#### Author

Robert Heller <heller@deepsoft.com>

### 9.79.2.2 ~LinuxGpio()

```
linuxgpio::LinuxGpio::~~LinuxGpio ( )
```

Destructor.

Unexport the pin.

## 9.79.3 Member Function Documentation

#### 9.79.3.1 Clr()

```
linuxgpio::LinuxGpio::Clr ( )
```

Set the pin to logic false.

#### 9.79.3.2 Get()

```
linuxgpio::LinuxGpio::Get ( )
```

Get the pin's logic state.

#### 9.79.3.3 is\_output()

```
linuxgpio::LinuxGpio::is_output ( )
```

Returns a boolean value indicating whether the pin is an output pin or not.

##### Returns

A boolean flag, true if this is an output, false if it is an input.

#### 9.79.3.4 read()

```
linuxgpio::LinuxGpio::read ( )
```

Read the value of the pin.

##### Returns

The value of the pin, 1 or 0.

#### 9.79.3.5 Set()

```
linuxgpio::LinuxGpio::Set ( )
```

Set the pin to logic true.

#### 9.79.3.6 write()

```
linuxgpio::LinuxGpio::write (
    value )
```

Write value to the pin.

#### Parameters

<i>value</i>	The value to write, either 1 or any non-zero value for high or 0 for low.
--------------	---

### 9.79.4 Member Data Documentation

#### 9.79.4.1 DIRECTIONFMT

linuxgpio::LinuxGpio::DIRECTIONFMT [static], [private]

The format string to generate the name of the direction control file.

#### 9.79.4.2 EXPORT

linuxgpio::LinuxGpio::EXPORT [static], [private]

The name of the export control file.

#### 9.79.4.3 UNEXPORT

linuxgpio::LinuxGpio::UNEXPORT [static], [private]

The name of the unexport control file.

#### 9.79.4.4 VALUEFMT

linuxgpio::LinuxGpio::VALUEFMT [static], [private]

The format string to generate the name of the value file.

## 9.80 xpressnet::LocomotiveAddress Class Reference

Locomotive address.

## Public Member Functions

- [LocomotiveAddress](#) (name, k, a)  
*Constructor.*
- [AddressType](#) ()  
*Return address type.*
- [Address](#) ()  
*Return address.*

## Private Attributes

- [\\_addressType](#)  
*Address type.*
- [\\_address](#)  
*Address.*

### 9.80.1 Detailed Description

Locomotive address.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.80.2 Constructor & Destructor Documentation

#### 9.80.2.1 LocomotiveAddress()

```
xpressnet::LocomotiveAddress::LocomotiveAddress (
    name ,
    k ,
    a )
```

Constructor.

#### Parameters

<i>k</i>	K (address type code).
<i>a</i>	Address.



### 9.80.3 Member Function Documentation

#### 9.80.3.1 Address()

```
xpressnet::LocomotiveAddress::Address ( )
```

Return address.

#### 9.80.3.2 AddressType()

```
xpressnet::LocomotiveAddress::AddressType ( )
```

Return address type.

### 9.80.4 Member Data Documentation

#### 9.80.4.1 \_address

```
xpressnet::LocomotiveAddress::_address [private]
```

Address.

#### 9.80.4.2 \_addressType

```
xpressnet::LocomotiveAddress::_addressType [private]
```

Address type.

## 9.81 CabWidgets::LocomotiveDirection Class Reference

Locomotive Direction widget.

## Public Member Functions

- [direction](#) ()  
*Return the current direction.*
- [LocomotiveDirection](#) (name,...)  
*Build and install all component widgets and process configuration.*
- [invoke](#) ()  
*Method to invoke the widget.*
- [direction\\_sense](#) (dir)  
*Method to set the sensed direction.*

## Private Member Functions

- [\\_setdirection](#) (dir)  
*Set the current direction.*

## Private Attributes

- [reverse](#)  
*Reverse button component.*
- [currentDirection](#)  
*Current direction label component.*
- [forward](#)  
*Forward button component.*
- [\\_direction](#)  
*The current direction.*

## Static Private Attributes

- static [\\_left](#)  
*Left bitmap (reverse button).*
- static [\\_right](#)  
*Right bitmap (forward button).*

### 9.81.1 Detailed Description

Locomotive Direction widget.

This widget implements Locomotive Direction control / display. There are buttons for selecting the direction and the current direction is displayed.

#### Parameters

<i>path</i>	Pathname of the widget.
...	Options:
	<ul style="list-style-type: none"> <li>• -command Script to call when the direction is changed. The new direction is appended</li> </ul>

**Author**

Robert Heller <heller@deepsoft.com>

**9.81.2 Constructor & Destructor Documentation****9.81.2.1 LocomotiveDirection()**

```
CabWidgets::LocomotiveDirection::LocomotiveDirection (
    name ,
    ... )
```

Build and install all component widgets and process configuration.

**Parameters**

...	Argument list (option value pairs). Gets passed to the implicitly defined configurelist method.
-----	---

**9.81.3 Member Function Documentation****9.81.3.1 \_setdirection()**

```
CabWidgets::LocomotiveDirection::_setdirection (
    dir ) [private]
```

Set the current direction.

Bound to direction buttons.

**Parameters**

<i>dir</i>	Localized string containing the direction.
------------	--

### 9.81.3.2 direction()

```
CabWidgets::LocomotiveDirection::direction ( )
```

Return the current direction.

### 9.81.3.3 direction\_sense()

```
CabWidgets::LocomotiveDirection::direction_sense (
    dir )
```

Method to set the sensed direction.

#### Parameters

<i>dir</i>	The localized direction to set.
------------	---------------------------------

### 9.81.3.4 invoke()

```
CabWidgets::LocomotiveDirection::invoke ( )
```

Method to invoke the widget.

This calls the script (if any) defined by the -command option.

## 9.81.4 Member Data Documentation

### 9.81.4.1 \_direction

```
CabWidgets::LocomotiveDirection::_direction [private]
```

The current direction.

#### 9.81.4.2 \_left

`CabWidgets::LocomotiveDirection::_left` [static], [private]

Left bitmap (reverse button).

#### 9.81.4.3 \_right

`CabWidgets::LocomotiveDirection::_right` [static], [private]

Right bitmap (forward button).

#### 9.81.4.4 currentDirection

`CabWidgets::LocomotiveDirection::currentDirection` [private]

Current direction label component.

#### 9.81.4.5 forward

`CabWidgets::LocomotiveDirection::forward` [private]

Forward button component.

#### 9.81.4.6 reverse

`CabWidgets::LocomotiveDirection::reverse` [private]

Reverse button component.

## 9.82 xpressnet::LocomotiveInformation Class Reference

Locomotive information.

## Public Member Functions

- [Address](#) ()  
*Return address.*
- [Available](#) ()  
*Return available flag.*
- [Direction](#) ()  
*Return direction.*
- [SpeedStepMode](#) ()  
*Return speed step mode.*
- [Speed](#) ()  
*Return speed.*
- [Function](#) (f)  
*Return function status.*
- [MTR](#) ()  
*Return Muti-unit address.*
- [Address2](#) ()  
*Return the address of second unit in double header.*
- [LocomotiveInformation](#) (name, a, avail=0, dir="", ssm="", s=0, f0=0, f1=0, f2=0, f3=0, f4=0, f5=0, f6=0, f7=0, f8=0, f9=0, f10=0, f11=0, f12=0, mtraddr=0, addr2=0xffff)  
*Constructor.*

## Private Attributes

- [\\_address](#)  
*Locomotive address.*
- [\\_available](#)  
*Locomotive is available.*
- [\\_direction](#)  
*Locomotive direction.*
- [\\_speedstep](#)  
*Locomotive speed step mode.*
- [\\_speed](#)  
*Locomotive speed.*
- [\\_function0](#)  
*Function 0.*
- [\\_function1](#)  
*Function 1.*
- [\\_function2](#)  
*Function 2.*
- [\\_function3](#)  
*Function 3.*
- [\\_function4](#)  
*Function 4.*
- [\\_function5](#)  
*Function 5.*
- [\\_function6](#)

- [\\_function7](#)  
*Function 6.*
- [\\_function8](#)  
*Function 7.*
- [\\_function9](#)  
*Function 8.*
- [\\_function10](#)  
*Function 9.*
- [\\_function11](#)  
*Function 10.*
- [\\_function12](#)  
*Function 11.*
- [\\_mtraddress](#)  
*Function 12.*
- [\\_address2](#)  
*Multi-unit address.*
- [\\_address2](#)  
*Double header address.*

### 9.82.1 Detailed Description

Locomotive information.

Author

Robert Heller <heller@deepsoft.com>

### 9.82.2 Constructor & Destructor Documentation

#### 9.82.2.1 LocomotiveInformation()

```
xpressnet::LocomotiveInformation::LocomotiveInformation (
    name ,
    a ,
    avail = 0,
    dir = "",
    ssm = "",
    s = 0,
    f0 = 0,
    f1 = 0,
    f2 = 0,
    f3 = 0,
    f4 = 0,
    f5 = 0,
    f6 = 0,
```

```

f7   = 0,
f8   = 0,
f9   = 0,
f10  = 0,
f11  = 0,
f12  = 0,
mtraddr = 0,
addr2 = 0xffff )

```

Constructor.

#### Parameters

<i>a</i>	Locomotive address.
<i>avail</i>	Available flag.
<i>dir</i>	Direction.
<i>ssm</i>	Speed step mode.
<i>s</i>	Locomotive speed.
<i>f0</i>	Function 0 status.
<i>f1</i>	Function 1 status.
<i>f2</i>	Function 2 status.
<i>f3</i>	Function 3 status.
<i>f4</i>	Function 4 status.
<i>f5</i>	Function 5 status.
<i>f6</i>	Function 6 status.
<i>f7</i>	Function 7 status.
<i>f8</i>	Function 8 status.
<i>f9</i>	Function 9 status.
<i>f10</i>	Function 10 status.
<i>f11</i>	Function 11 status.
<i>f12</i>	Function 12 status.
<i>mtraddr</i>	MTR address.
<i>addr2</i>	Double header address.

### 9.82.3 Member Function Documentation

#### 9.82.3.1 Address()

```
xpressnet::LocomotiveInformation::Address ( )
```

Return address.



### 9.82.3.2 Address2()

```
xpressnet::LocomotiveInformation::Address2 ( )
```

Return the address of second unit in double header.

### 9.82.3.3 Available()

```
xpressnet::LocomotiveInformation::Available ( )
```

Return available flag.

### 9.82.3.4 Direction()

```
xpressnet::LocomotiveInformation::Direction ( )
```

Return direction.

### 9.82.3.5 Function()

```
xpressnet::LocomotiveInformation::Function (
    f )
```

Return function status.

#### Parameters

<i>f</i>	Function whose status to return.
----------	----------------------------------

### 9.82.3.6 MTR()

```
xpressnet::LocomotiveInformation::MTR ( )
```

Return Muti-unit address.

#### 9.82.3.7 Speed()

```
xpressnet::LocomotiveInformation::Speed ( )
```

Return speed.

#### 9.82.3.8 SpeedStepMode()

```
xpressnet::LocomotiveInformation::SpeedStepMode ( )
```

Return speed step mode.

### 9.82.4 Member Data Documentation

#### 9.82.4.1 \_address

```
xpressnet::LocomotiveInformation::_address [private]
```

Locomotive address.

#### 9.82.4.2 \_address2

```
xpressnet::LocomotiveInformation::_address2 [private]
```

Double header address.

#### 9.82.4.3 \_available

```
xpressnet::LocomotiveInformation::_available [private]
```

Locomotive is available.

#### 9.82.4.4 `_direction`

`xpressnet::LocomotiveInformation::_direction` [private]

Locomotive direction.

#### 9.82.4.5 `_function0`

`xpressnet::LocomotiveInformation::_function0` [private]

Function 0.

#### 9.82.4.6 `_function1`

`xpressnet::LocomotiveInformation::_function1` [private]

Function 1.

#### 9.82.4.7 `_function10`

`xpressnet::LocomotiveInformation::_function10` [private]

Function 10.

#### 9.82.4.8 `_function11`

`xpressnet::LocomotiveInformation::_function11` [private]

Function 11.

#### 9.82.4.9 `_function12`

`xpressnet::LocomotiveInformation::_function12` [private]

Function 12.

**9.82.4.10 \_function2**

```
xpressnet::LocomotiveInformation::_function2 [private]
```

Function 2.

**9.82.4.11 \_function3**

```
xpressnet::LocomotiveInformation::_function3 [private]
```

Function 3.

**9.82.4.12 \_function4**

```
xpressnet::LocomotiveInformation::_function4 [private]
```

Function 4.

**9.82.4.13 \_function5**

```
xpressnet::LocomotiveInformation::_function5 [private]
```

Function 5.

**9.82.4.14 \_function6**

```
xpressnet::LocomotiveInformation::_function6 [private]
```

Function 6.

**9.82.4.15 \_function7**

```
xpressnet::LocomotiveInformation::_function7 [private]
```

Function 7.

#### 9.82.4.16 \_function8

xpressnet::LocomotiveInformation::\_function8 [private]

Function 8.

#### 9.82.4.17 \_function9

xpressnet::LocomotiveInformation::\_function9 [private]

Function 9.

#### 9.82.4.18 \_mtraddress

xpressnet::LocomotiveInformation::\_mtraddress [private]

Multi-unit address.

#### 9.82.4.19 \_speed

xpressnet::LocomotiveInformation::\_speed [private]

Locomotive speed.

#### 9.82.4.20 \_speedstep

xpressnet::LocomotiveInformation::\_speedstep [private]

Locomotive speed step mode.

## 9.83 CabWidgets::LocomotiveSpeed Class Reference

Locomotive Speed widget.

## Public Member Functions

- [speed](#) ()  
*Method to return the current speed.*
- [setspeed](#) (speed)  
*Method to set the sensed speed.*
- [LocomotiveSpeed](#) (name,...)  
*Build and install all component widgets and process configuration.*
- [invoke](#) ()  
*Method to invoke the widget.*

## Private Member Functions

- [\\_setspeed](#) (newspeed)  
*Set the speed, bound to the bar -command option.*
- [\\_stop](#) ()  
*Stop method, bound to the stop button.*
- [\\_up1](#) ()  
*Up by one method, bound to the slow up button.*
- [\\_up10](#) ()  
*Up by 10 method, bound to the fast up button.*
- [\\_down1](#) ()  
*Down by one method, bound to the slow down button.*
- [\\_down10](#) ()  
*Down by one method, bound to the fast down button.*

## Private Attributes

- [leftbuttons](#)  
*Left buttons component (small increments).*
- [up1](#)  
*Up by one button.*
- [down1](#)  
*Down by one button.*
- [rightbuttons](#)  
*Right buttons component (larger increments).*
- [up10](#)  
*Up by ten button.*
- [down10](#)  
*Down by one button.*
- [bar](#)  
*Current speed bar.*
- [stop](#)  
*Stop button.*
- [\\_speed](#)  
*The current speed.*

## Static Private Attributes

- static [\\_up](#)  
*Bitmap for up button.*
- static [\\_down](#)  
*Bitmap for down button.*
- static [\\_up10](#)  
*Bitmap for fast up button.*
- static [\\_down10](#)  
*Bitmap for fast down button.*
- static [\\_stop](#)  
*Bitmap for fast down button.*

### 9.83.1 Detailed Description

Locomotive Speed widget.

This widget implements Locomotive Speed control / display. There are buttons for increasing or decreasing speed either by units of 1 or by units of 10. Plus there is a bar showing the current relative speed.

#### Parameters

<i>path</i>	Pathname of the widget.
...	Options: <ul style="list-style-type: none"> <li>• -command Script to call when the speed is changed. The new speed is appended.</li> </ul>

#### Author

Robert Heller <heller@deepsoft.com>

### 9.83.2 Constructor & Destructor Documentation

#### 9.83.2.1 LocomotiveSpeed()

```
CabWidgets::LocomotiveSpeed::LocomotiveSpeed (
    name ,
    ... )
```

Build and install all component widgets and process configuration.

**Parameters**

...	Argument list (option value pairs). Gets passed to the implicitly defined configurelist method.
-----	---

**9.83.3 Member Function Documentation****9.83.3.1 \_down1()**

```
CabWidgets::LocomotiveSpeed::_down1 ( ) [private]
```

Down by one method, bound to the slow down button.

**9.83.3.2 \_down10()**

```
CabWidgets::LocomotiveSpeed::_down10 ( ) [private]
```

Down by one method, bound to the fast down button.

**9.83.3.3 \_setspeed()**

```
CabWidgets::LocomotiveSpeed::_setspeed (
    newspeed ) [private]
```

Set the speed, bound to the bar -command option.

**9.83.3.4 \_stop()**

```
CabWidgets::LocomotiveSpeed::_stop ( ) [private]
```

Stop method, bound to the stop button.



#### 9.83.3.5 \_up1()

```
CabWidgets::LocomotiveSpeed::_up1 ( ) [private]
```

Up by one method, bound to the slow up button.

#### 9.83.3.6 \_up10()

```
CabWidgets::LocomotiveSpeed::_up10 ( ) [private]
```

Up by 10 method, bound to the fast up button.

#### 9.83.3.7 invoke()

```
CabWidgets::LocomotiveSpeed::invoke ( )
```

Method to invoke the widget.

This calls the script (if any) defined by the -command option.

#### 9.83.3.8 setspeed()

```
CabWidgets::LocomotiveSpeed::setspeed (
    speed )
```

Method to set the sensed speed.

#### 9.83.3.9 speed()

```
CabWidgets::LocomotiveSpeed::speed ( )
```

Method to return the current speed.

### 9.83.4 Member Data Documentation

#### 9.83.4.1 `_down`

`CabWidgets::LocomotiveSpeed::_down` [static], [private]

Bitmap for down button.

#### 9.83.4.2 `_down10`

`CabWidgets::LocomotiveSpeed::_down10` [static], [private]

Bitmap for fast down button.

#### 9.83.4.3 `_speed`

`CabWidgets::LocomotiveSpeed::_speed` [private]

The current speed.

#### 9.83.4.4 `_stop`

`CabWidgets::LocomotiveSpeed::_stop` [static], [private]

Bitmap for fast down button.

#### 9.83.4.5 `_up`

`CabWidgets::LocomotiveSpeed::_up` [static], [private]

Bitmap for up button.

#### 9.83.4.6 `_up10`

`CabWidgets::LocomotiveSpeed::_up10` [static], [private]

Bitmap for fast up button.

#### 9.83.4.7 bar

`CabWidgets::LocomotiveSpeed::bar` [private]

Current speed bar.

#### 9.83.4.8 down1

`CabWidgets::LocomotiveSpeed::down1` [private]

Down by one button.

#### 9.83.4.9 down10

`CabWidgets::LocomotiveSpeed::down10` [private]

Down by one button.

#### 9.83.4.10 leftbuttons

`CabWidgets::LocomotiveSpeed::leftbuttons` [private]

Left buttons component (small increments).

#### 9.83.4.11 rightbuttons

`CabWidgets::LocomotiveSpeed::rightbuttons` [private]

Right buttons component (larger increments).

#### 9.83.4.12 stop

`CabWidgets::LocomotiveSpeed::stop` [private]

Stop button.

#### 9.83.4.13 up1

```
CabWidgets::LocomotiveSpeed::up1 [private]
```

Up by one button.

#### 9.83.4.14 up10

```
CabWidgets::LocomotiveSpeed::up10 [private]
```

Up by ten button.

## 9.84 FCFSupport::LogMessageCallback Class Reference

A callback to log a message.

```
#include <CallBack.h>
```

### Public Types

- enum [MessageType](#) { [Infomational](#) =1, [Warning](#) =2, [Error](#) =3 }  
*The three types of messages.*

### Public Member Functions

- [LogMessageCallback](#) ()  
*Constructor.*
- virtual [~LogMessageCallback](#) ()  
*Destructor.*
- virtual void [LogMessage](#) ([MessageType](#) Type, const string Message) const  
*Log message callback function.*

#### 9.84.1 Detailed Description

A callback to log a message.

This callback class is used to display various sorts of messages in an application dependent way. There are three types of messages, informational messages, warning messages, and error messages. Infomational messages are just to inform the user of of important things that are happening. Warning messages are to inform the user of minor, correctable, problems. Error are to inform the user of serious problems that need to be fixed before proceeding much further.

```
@author Robert Heller \<heller\@deepsoft.com\>
```

## 9.84.2 Member Enumeration Documentation

### 9.84.2.1 MessageType

enum `FCFSupport::LogMessageCallback::MessageType`

The three types of messages.

### Enumerator

Informational	Random informational messages.
Warning	Warning messages.
Error	Error messages.

## 9.84.3 Constructor & Destructor Documentation

### 9.84.3.1 LogMessageCallback()

```
FCFSupport::LogMessageCallback::LogMessageCallback ( ) [inline]
```

Constructor.

The base constructor does nothing. It is presumed that a derived class might do something useful.

### 9.84.3.2 ~LogMessageCallback()

```
virtual FCFSupport::LogMessageCallback::~~LogMessageCallback ( ) [inline], [virtual]
```

Destructor.

The base destructor does nothing. It is presumed that a derived class might do something useful.

## 9.84.4 Member Function Documentation

### 9.84.4.1 LogMessage()

```
virtual void FCFSupport::LogMessageCallback::LogMessage (
    MessageType Type,
    const string Message ) const [inline], [virtual]
```

Log message callback function.

Display a specific type of message in an application specific way.

## Parameters

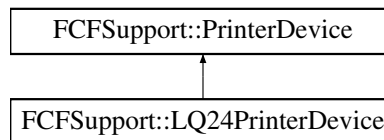
<i>Type</i>	The message type.
<i>Message</i>	The message itself.

## 9.85 FCFSupport::LQ24PrinterDevice Class Reference

Class for an LQ24 compatible printer.

```
#include <LQ24Printer.h>
```

Inheritance diagram for FCFSupport::LQ24PrinterDevice:



### Public Member Functions

- [LQ24PrinterDevice](#) (const string filename="", const string title="", [PageSize pageSize=Letter](#), char \*\*outmessage=NULL)  
*Constructor.*
- virtual bool [OpenPrinter](#) (const string filename, [PageSize pageSize=Letter](#), char \*\*outmessage=NULL)  
*Member function to open the printer.*
- virtual bool [ClosePrinter](#) (char \*\*outmessage)  
*Close the printer.*
- virtual bool [SetTypeSpacing](#) ([TypeSpacing](#) spacing)  
*Set the the spacing.*
- virtual bool [SetTypeWeight](#) ([TypeWeight](#) weight)  
*Set the type weight.*
- virtual bool [SetTypeSlant](#) ([TypeSlant](#) slant)  
*Set the type slant.*
- virtual bool [NewPage](#) (const string heading="")  
*Perform a page feed and print a heading.*
- virtual bool [PutLine](#) (const string line)  
*Print out a string and follow it with a new line sequence.*
- virtual bool [Put](#) (const string text)  
*Print a string of text.*
- virtual bool [Tab](#) (int column)  
*Tab over to the specified column.*
- virtual [~LQ24PrinterDevice](#) ()  
*Destructor.*

## Private Types

- enum `ChCodes` { `FF` = 12, `SI` = 15, `DC2` = 18, `ESC` = 27 }  
*Special character codes.*

## Private Attributes

- ofstream `printerStream`  
*Output stream.*
- int `currentColumn`  
*Current column.*
- double `currentColumnFraction`  
*Current column fraction.*
- `TypeSpacing` `currentSpacing`  
*Current spacing.*
- `TypeWeight` `currentWeight`  
*Current weight.*
- `TypeSlant` `currentSlant`  
*Current slant.*
- double `oneColumnWidthFraction`  
*One column's width fraction.*

## Additional Inherited Members

### 9.85.1 Detailed Description

Class for an LQ24 compatible printer.

This is Epson's 24-bit dot matrix printers.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.85.2 Member Enumeration Documentation

#### 9.85.2.1 ChCodes

```
enum FCFSupport::LQ24PrinterDevice::ChCodes [private]
```

Special character codes.

These character codes introduce various special printer functions and modes.



## Enumerator

FF	Form feed. This code causes a page feed.
SI	Shift In This character starts condensed (half width) spacing.
DC2	Device control 2. This character ends condensed (half width) spacing.
ESC	Escape. This character is used to introduce a number of escape sequences to perform a number of printer functions and/or set various printing modes.

## 9.85.3 Constructor &amp; Destructor Documentation

## 9.85.3.1 LQ24PrinterDevice()

```
FCFSupport::LQ24PrinterDevice::LQ24PrinterDevice (
    const string filename = "",
    const string title = "",
    PageSize pageSize = Letter,
    char ** outmessage = NULL )
```

Constructor.

Create a new printer device instance from a set of parameters, all of which have default values, so this also doubles as the default base constructor.

## Parameters

<i>filename</i>	Output filename.
<i>title</i>	An internal document title string.
<i>pageSize</i>	The page size to use. This parameter is not used.
<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur. This parameter is hidden from the Tcl interface.

## 9.85.3.2 ~LQ24PrinterDevice()

```
virtual FCFSupport::LQ24PrinterDevice::~LQ24PrinterDevice ( ) [virtual]
```

Destructor.

Close the printer.

## 9.85.4 Member Function Documentation

### 9.85.4.1 ClosePrinter()

```
virtual bool FCFSupport::LQ24PrinterDevice::ClosePrinter (
    char ** outmessage ) [virtual]
```

Close the printer.

#### Parameters

<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur. This parameter is hidden from the Tcl interface.
-------------------	--

Reimplemented from [FCFSupport::PrinterDevice](#).

### 9.85.4.2 NewPage()

```
virtual bool FCFSupport::LQ24PrinterDevice::NewPage (
    const string heading = "" ) [virtual]
```

Perform a page feed and print a heading.

#### Parameters

<i>heading</i>	The heading string.
----------------	---------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

### 9.85.4.3 OpenPrinter()

```
virtual bool FCFSupport::LQ24PrinterDevice::OpenPrinter (
    const string filename,
    PageSize pageSize = Letter,
    char ** outmessage = NULL ) [virtual]
```

Member function to open the printer.

## Parameters

<i>filename</i>	Output filename.
<i>pageSize</i>	The page size to use.
<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur. This parameter is hidden from the Tcl interface.

Reimplemented from [FCFSupport::PrinterDevice](#).

## 9.85.4.4 Put()

```
virtual bool FCFSupport::LQ24PrinterDevice::Put (  
    const string text ) [virtual]
```

Print a string of text.

Don't include a newline.

## Parameters

<i>text</i>	The string to print.
-------------	----------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

## 9.85.4.5 PutLine()

```
virtual bool FCFSupport::LQ24PrinterDevice::PutLine (  
    const string line ) [virtual]
```

Print out a string and follow it with a new line sequence.

## Parameters

<i>line</i>	The line to print.
-------------	--------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

#### 9.85.4.6 SetTypeSlant()

```
virtual bool FCFSupport::LQ24PrinterDevice::SetTypeSlant (
    TypeSlant slant ) [virtual]
```

Set the type slant.

##### Parameters

<i>slant</i>	The new type slant.
--------------	---------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

#### 9.85.4.7 SetTypeSpacing()

```
virtual bool FCFSupport::LQ24PrinterDevice::SetTypeSpacing (
    TypeSpacing spacing ) [virtual]
```

Set the the spacing.

##### Parameters

<i>spacing</i>	The new type spacing.
----------------	-----------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

#### 9.85.4.8 SetTypeWeight()

```
virtual bool FCFSupport::LQ24PrinterDevice::SetTypeWeight (
    TypeWeight weight ) [virtual]
```

Set the type weight.

##### Parameters

<i>weight</i>	The new type weight.
---------------	----------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

#### 9.85.4.9 Tab()

```
virtual bool FCFSupport::LQ24PrinterDevice::Tab (  
    int column ) [virtual]
```

Tab over to the specified column.

##### Parameters

<i>column</i>	The desired tab column.
---------------	-------------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

### 9.85.5 Member Data Documentation

#### 9.85.5.1 currentColumn

```
int FCFSupport::LQ24PrinterDevice::currentColumn [private]
```

Current column.

#### 9.85.5.2 currentColumnFraction

```
double FCFSupport::LQ24PrinterDevice::currentColumnFraction [private]
```

Current column fraction.

#### 9.85.5.3 currentSlant

```
TypeSlant FCFSupport::LQ24PrinterDevice::currentSlant [private]
```

Current slant.

#### 9.85.5.4 currentSpacing

`TypeSpacing FCFSupport::LQ24PrinterDevice::currentSpacing [private]`

Current spacing.

#### 9.85.5.5 currentWeight

`TypeWeight FCFSupport::LQ24PrinterDevice::currentWeight [private]`

Current weight.

#### 9.85.5.6 oneColumnWidthFraction

`double FCFSupport::LQ24PrinterDevice::oneColumnWidthFraction [private]`

One column's width fraction.

#### 9.85.5.7 printerStream

`ofstream FCFSupport::LQ24PrinterDevice::printerStream [private]`

Output stream.

## 9.86 mainwindow Class Reference

A widget that is heavily extended from the BWidget MainFrame windget.

## Public Member Functions

- [buttons\\_add](#) (...)   
*Method to add a button to the button menu.*
- [buttons\\_delete](#) (index)   
*Method to delete a button from the button menu.*
- [buttons\\_insert](#) (index,...)   
*Method to insert a button to the button menu.*
- [buttons\\_itemconfigure](#) (index,...)   
*Method to configure a button in the button menu.*
- [buttons\\_hide](#) ()   
*Method to hide the button menu.*
- [buttons\\_show](#) ()   
*Method to show the button menu.*
- [slideout\\_add](#) (name)   
*Method to add a new slideout frame to the main window.*
- [slideout\\_show](#) (name)   
*Method to show (display) a slideout frame.*
- [slideout\\_hide](#) (name)   
*Method to hide a slideout frame.*
- [slideout\\_getframe](#) (name)   
*Method to get the frame of a slideout frame.*
- [slideout\\_issownp](#) (name)   
*Method to test to see if the named slideout is being shown.*
- [slideout\\_list](#) ()   
*Method to return a list of defined slideout frames.*
- [slideout\\_reqwidth](#) (name)   
*Method to return the requested width of the named slideout frame.*
- [toolbar\\_add](#) (name)   
*Method to add a toolbar to the main frame.*
- [toolbar\\_show](#) (name)   
*Method to show a toolbar.*
- [toolbar\\_hide](#) (name)   
*Method to hide a toolbar.*
- [toolbar\\_setbuttonstate](#) (name, state)   
*Method to set the state of the buttons in a toolbar.*
- [toolbar\\_addbutton](#) (name, bname,...)   
*Method to add a button to a toolbar.*
- [toolbar\\_buttonconfigure](#) (name, bname,...)   
*Method to configure a button on a toolbar.*
- [toolbar\\_buttoncget](#) (name, bname, option)   
*Method to get a configuration option of a button on a toolbar.*
- [menu\\_activate](#) (menuid, index)   
*Method to activate a menu on the main frame.*
- [menu\\_add](#) (menuid, entrytype,...)   
*Method to add a menu entry to a menu on the main frame.*
- [menu\\_delete](#) (menuid, index,...)

*Method to delete an entry from a menu.*

- [menu\\_entrycget](#) (menuid, index, option)

*Method to get an option value of a menu entry.*

- [menu\\_entryconfigure](#) (menuid, index,...)

*Method to configure options of a menu entry.*

- [menu\\_sethelpvar](#) (menuid)

*Method to set the bind the status line to the help variable of the menu.*

- [menu\\_index](#) (menuid, index)

*Method to get the index of a menu entry.*

- [menu\\_insert](#) (menuid, index, entrytype,...)

*Method to insert a menu entry to a menu on the main frame.*

- [menu\\_invoke](#) (menuid, index)

*Method to invoke a menu entry.*

- [menu\\_type](#) (menuid, index)

*Method to return the type of a menu entry.*

- [showit](#) (extraX=0)

*Method to show the main window.*

- [setstatus](#) (statusmessage)

*Method to set the status message.*

- [setprogress](#) (progressvalue)

*Method to set the progress bar value.*

- [mainwindow](#) (name,...)

*Constructor build a full featured main window.*

## Private Attributes

- [scrollwindow](#)

*ScrollWindow component.*

- [wipmessage](#)

*Work-In-Progress message component.*

- [buttons](#)

*Right button box.*

- [panewindow](#)

*PaneWindow holding the main area and the right widgets (command buttons and the slideouts).*

- [slideouts](#)

*Slideout map.*

- [toolbars](#)

*Toolbar map.*

- [numtoolbars](#)

*The number of toolbars.*

- [progress](#)

*The value of the progreee bar.*

- [status](#)

*Status value.*



### 9.86.1 Detailed Description

A widget that is heavily extended from the BWidget MainFrame widget.

This widget starts with a MainFrame, and adds a paned window with a scrolled window and a button menu, and zero or more slide out frames. also management methods for toolbars and for menus.

## Parameters

<i>path</i>	The widget path.
...	Options: <ul style="list-style-type: none"> <li>• <code>-menu</code> The basic MainFrame <code>-menu</code> option. Defaults to the Motif standard set of menus (File, Edit, View, Options, and Help).</li> <li>• <code>-extramenus</code> Like the basic MainFrame <code>-menu</code> option, but can be used when the just additional menus need to be added to the standard set.</li> <li>• <code>-height</code> Widget height. Delegated to the hull (MainFrame) widget.</li> <li>• <code>-width</code> Widget width. Delegated to the hull (MainFrame) widget.</li> <li>• <code>-separator</code> Include a separator between windows on the MainFrame widget.</li> <li>• <code>-dontwithdraw</code> Boolean to suppress withdrawing the toplevel while it is being built.</li> <li>• <code>-scrolling</code> Boolean to enable the scrollwindow.</li> </ul>

## Author

Robert Heller <heller@deepsoft.com>

## 9.86.2 Package provided

MainWindow 1.0

## 9.86.3 Constructor & Destructor Documentation

### 9.86.3.1 mainwindow()

```
mainwindow::mainwindow (
    name ,
    ... )
```

Constructor build a full featured main window.

## Parameters

...	Option value list.
-----	--------------------

## 9.86.4 Member Function Documentation

### 9.86.4.1 buttons\_add()

```
mainwindow::buttons_add (
    ... )
```

Method to add a button to the button menu.

See the ButtonBox add method.

#### Parameters

...	Arguments passed to the ButtonBox add method.
-----	---

### 9.86.4.2 buttons\_delete()

```
mainwindow::buttons_delete (
    index )
```

Method to delete a button from the button menu.

See the ButtonBox delete method.

#### Parameters

<i>index</i>	Passed to the ButtonBox delete method.
--------------	--

### 9.86.4.3 buttons\_hide()

```
mainwindow::buttons_hide ( )
```

Method to hide the button menu.

#### 9.86.4.4 buttons\_insert()

```
mainwindow::buttons_insert (
    index ,
    ... )
```

Method to insert a button to the button menu.

See the ButtonBox insert method.

##### Parameters

...	Arguments passed to the ButtonBox insert method.
-----	--

#### 9.86.4.5 buttons\_itemconfigure()

```
mainwindow::buttons_itemconfigure (
    index ,
    ... )
```

Method to configure a button in the button menu.

See the ButtonBox itemconfigure method.

##### Parameters

<i>index</i>	Argument passed to the ButtonBox itemconfigure method.
...	Arguments passed to the ButtonBox itemconfigure method.

#### 9.86.4.6 buttons\_show()

```
mainwindow::buttons_show ( )
```

Method to show the button menu.

#### 9.86.4.7 menu\_activate()

```
mainwindow::menu_activate (
    menuid ,
    index )
```

Method to activate a menu on the main frame.

## Parameters

<i>menuid</i>	Menu id.
<i>index</i>	Menu item index.

## 9.86.4.8 menu\_add()

```
mainwindow::menu_add (
    menuid ,
    entrytype ,
    ... )
```

Method to add a menu entry to a menu on the main frame.

## Parameters

<i>menuid</i>	Menu id.
<i>entrytype</i>	The type of entry.
...	The arguments to pass to the entry creation command.

## 9.86.4.9 menu\_delete()

```
mainwindow::menu_delete (
    menuid ,
    index ,
    ... )
```

Method to delete an entry from a menu.

## Parameters

<i>menuid</i>	Menu id.
<i>index</i>	Menu item index.
...	The arguments to pass to the menu delete command.

## 9.86.4.10 menu\_entrycget()

```
mainwindow::menu_entrycget (
    menuid ,
```

```
index ,  
option )
```

Method to get an option value of a menu entry.

#### Parameters

<i>menuid</i>	Menu id.
<i>index</i>	Menu item index.
<i>option</i>	The option to fetch.

#### 9.86.4.11 menu\_entryconfigure()

```
mainwindow::menu_entryconfigure (   
    menuid ,  
    index ,  
    ... )
```

Method to configure options of a menu entry.

#### Parameters

<i>menuid</i>	Menu id.
<i>index</i>	Menu item index.
...	The arguments to pass on to entryconfigure.

#### 9.86.4.12 menu\_index()

```
mainwindow::menu_index (   
    menuid ,  
    index )
```

Method to get the index of a menu entry.

#### Parameters

<i>menuid</i>	Menu id.
<i>index</i>	The index of the menu entry.

#### 9.86.4.13 menu\_insert()

```
mainwindow::menu_insert (
    menuid ,
    index ,
    entrytype ,
    ... )
```

Method to insert a menu entry to a menu on the main frame.

##### Parameters

<i>menuid</i>	Menu id.
<i>index</i>	The index to insert before.
<i>entrytype</i>	The type of entry.
...	The arguments to pass to the entry creation command.

#### 9.86.4.14 menu\_invoke()

```
mainwindow::menu_invoke (
    menuid ,
    index )
```

Method to invoke a menu entry.

##### Parameters

<i>menuid</i>	Menu id.
<i>index</i>	The index to invoke.

#### 9.86.4.15 menu\_sethelpvar()

```
mainwindow::menu_sethelpvar (
    menuid )
```

Method to set the bind the status line to the help variable of the menu.

##### Parameters

<i>menuid</i>	Menu id.
---------------	----------

**9.86.4.16 menu\_type()**

```
mainwindow::menu_type (
    menuid ,
    index )
```

Method to return the type of a menu entry.

**Parameters**

<i>menuid</i>	Menu id.
<i>index</i>	The index to get the type of.

**9.86.4.17 setprogress()**

```
mainwindow::setprogress (
    progressvalue )
```

Method to set the progress bar value.

**Parameters**

<i>progressvalue</i>	The amount of the progress.
----------------------	-----------------------------

**9.86.4.18 setstatus()**

```
mainwindow::setstatus (
    statusmessage )
```

Method to set the status message.

**Parameters**

<i>statusmessage</i>	The status message to display.
----------------------	--------------------------------

**9.86.4.19 showit()**

```
mainwindow::showit (
    extraX = 0 )
```



Method to show the main window.

#### Parameters

<i>extraX</i>	Extra width to add when computing the position to map the window at. Defaults to 0.
---------------	---

#### 9.86.4.20 slideout\_add()

```
mainwindow::slideout_add (  
    name )
```

Method to add a new slideout frame to the main window.

A slide out frame is a frame that can be packed and unpacked as needed and is shown in the right pane of the pane window.

#### Parameters

<i>name</i>	The name of the slideout frame.
-------------	---------------------------------

#### 9.86.4.21 slideout\_getframe()

```
mainwindow::slideout_getframe (  
    name )
```

Method to get the frame of a slideout frame.

#### Parameters

<i>name</i>	The name of the slideout.
-------------	---------------------------

#### 9.86.4.22 slideout\_hide()

```
mainwindow::slideout_hide (  
    name )
```

Method to hide a slideout frame.

**Parameters**

<i>name</i>	The name of the slideout.
-------------	---------------------------

**9.86.4.23 slideout\_isshownp()**

```
mainwindow::slideout_isshownp (  
    name )
```

Method to test to see if the named slideout is being shown.

**Parameters**

<i>name</i>	The name of the slideout.
-------------	---------------------------

**9.86.4.24 slideout\_list()**

```
mainwindow::slideout_list ( )
```

Method to return a list of defined slideout frames.

**9.86.4.25 slideout\_reqwidth()**

```
mainwindow::slideout_reqwidth (  
    name )
```

Method to return the requested width of the named slideout frame.

**Parameters**

<i>name</i>	The name of the slideout.
-------------	---------------------------

**9.86.4.26 slideout\_show()**

```
mainwindow::slideout_show (  
    name )
```

Method to show (display) a slideout frame.

#### Parameters

<i>name</i>	The name of the slideout.
-------------	---------------------------

#### 9.86.4.27 toolbar\_add()

```
mainwindow::toolbar_add (  
    name )
```

Method to add a toolbar to the main frame.

#### Parameters

<i>name</i>	The name of the new toolbar.
-------------	------------------------------

#### 9.86.4.28 toolbar\_addbutton()

```
mainwindow::toolbar_addbutton (  
    name ,  
    bname ,  
    ... )
```

Method to add a button to a toolbar.

#### Parameters

<i>name</i>	The name of the toolbar.
<i>bname</i>	The name of the button.
...	Button configuration options (passed to Button).

#### 9.86.4.29 toolbar\_buttoncget()

```
mainwindow::toolbar_buttoncget (  
    name ,  
    bname ,  
    option )
```

Method to get a configuration option of a button on a toolbar.

**Parameters**

<i>name</i>	The name of the toolbar.
<i>bname</i>	The name of the button.
<i>option</i>	Button configuration option (passed to cget).

**9.86.4.30 toolbar\_buttonconfigure()**

```
mainwindow::toolbar_buttonconfigure (
    name ,
    bname ,
    ... )
```

Method to configure a button on a toolbar.

**Parameters**

<i>name</i>	The name of the toolbar.
<i>bname</i>	The name of the button.
...	Button configuration options (passed to configure).

**9.86.4.31 toolbar\_hide()**

```
mainwindow::toolbar_hide (
    name )
```

Method to hide a toolbar.

**Parameters**

<i>name</i>	The name of the toolbar.
-------------	--------------------------

**9.86.4.32 toolbar\_setbuttonstate()**

```
mainwindow::toolbar_setbuttonstate (
    name ,
    state )
```

Method to set the state of the buttons in a toolbar.

## Parameters

<i>name</i>	The name of the toolbar.
-------------	--------------------------

## 9.86.4.33 toolbar\_show()

```
mainwindow::toolbar_show (  
    name )
```

Method to show a toolbar.

## Parameters

<i>name</i>	The name of the toolbar.
-------------	--------------------------

## 9.86.5 Member Data Documentation

## 9.86.5.1 buttons

```
mainwindow::buttons [private]
```

Right button box.

## 9.86.5.2 numtoolbars

```
mainwindow::numtoolbars [private]
```

The number of toolbars.

## 9.86.5.3 panewindow

```
mainwindow::panewindow [private]
```

PaneWindow holding the main area and the right widgets (command buttons and the slideouts).

#### 9.86.5.4 progress

`mainwindow::progress` [private]

The value of the progreee bar.

#### 9.86.5.5 scrollwindow

`mainwindow::scrollwindow` [private]

ScrollWindow component.

#### 9.86.5.6 slideouts

`mainwindow::slideouts` [private]

Slideout map.

#### 9.86.5.7 status

`mainwindow::status` [private]

Status value.

#### 9.86.5.8 toolbars

`mainwindow::toolbars` [private]

Toolbar map.

#### 9.86.5.9 wipmessage

`mainwindow::wipmessage` [private]

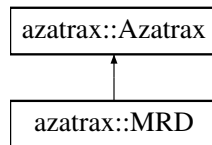
Work-In-Progress message component.

## 9.87 azatrax::MRD Class Reference

MRD I/O Class.

```
#include <mrd.h>
```

Inheritance diagram for azatrax::MRD:



### Classes

- union `status1_union`  
*Status byte 1 union type.*
- union `status2_union`  
*Status byte 2 union type.*

### Public Types

- enum `OperatingMode_Type` { `NonTurnoutSeparate` =0x31, `NonTurnoutDirectionSensing` =0x32, `TurnoutSolenoid` =0x34, `TurnoutMotor` =0x37 }  
*Operating Mode codes.*

### Public Member Functions

- `~MRD ()`  
*Base destructor.*
- `ErrorCode SetChan1 () const`  
*Set channel 1 relays and status bits.*
- `ErrorCode SetChan2 () const`  
*Set channel 2 relays and status bits.*
- `ErrorCode ClearExternallyChanged () const`  
*Clear 'ExternallyChanged' status bit.*
- `ErrorCode DisableExternal () const`  
*Disable external changes of turnout state.*
- `ErrorCode EnableExternal () const`  
*Enable external changes of turnout state.*
- `ErrorCode Identify_2 () const`  
*Identify 2.*
- `ErrorCode Identify_1_2 () const`  
*Identify 1 and 2.*

- [ErrorCode ResetStopwatch](#) () const  
*Reset Stopwatch.*
- bool [Sense\\_1](#) () const  
*Sensor one active.*
- bool [Sense\\_2](#) () const  
*Sensor two active.*
- bool [Latch\\_1](#) () const  
*Latch one.*
- bool [Latch\\_2](#) () const  
*Latch two.*
- bool [HasRelays](#) () const  
*Has Relays?*
- bool [ResetStatus](#) () const  
*Reset status?*
- bool [StopwatchTicking](#) () const  
*Stopwatch Ticking?*
- bool [ExternallyChanged](#) () const  
*Externally changed?*
- bool [AllowingExternalChanges](#) () const  
*Allowing External Changes?*
- [OperatingMode\\_Type OperatingMode](#) () const  
*Operating mode.*
- void [Stopwatch](#) (uint8\_t &fract, uint8\_t &seconds, uint8\_t &minutes, uint8\_t &hours) const  
*Stopwatch time.*

### Protected Member Functions

- [MRD](#) (const char \*serialnumber, char \*\*outmessage=NULL)  
*Base constructor.*

### Friends

- class [Azatrax](#)

### Additional Inherited Members

#### 9.87.1 Detailed Description

[MRD](#) I/O Class.

[MRD](#) interface class.

This class implements the interface logic for a MRD2-S or MRD2-U device.

The constructor opens a connection to a MRD2-S or MRD2-U device, given its serial number. Each MRD2-S or MRD2-U device has a unique, factory defined serial number, which is printed on a sticker attached to the module. This serial number is much like the MAC address of an Ethernet interface. The destructor closes the connection to the device and frees any resources allocated.

The class provides methods to send commands to the device, read back its state and interrogate the state read back. This way each class instance encapsulates each device instance.



## Author

Robert Heller <heller@deepsoft.com>

## 9.87.2 Member Enumeration Documentation

### 9.87.2.1 OperatingMode\_Type

```
enum azatrax::MRD::OperatingMode_Type
```

Operating Mode codes.

## Enumerator

NonTurnoutSeparate	Non Turnout, separate (-U always reports this).
NonTurnoutDirectionSensing	Non Turnout, Direction Sensing.
TurnoutSolenoid	Turnout, Solenoid (momentary action)
TurnoutMotor	Turnout, Motor (sustained action)

## 9.87.3 Constructor & Destructor Documentation

### 9.87.3.1 MRD()

```
azatrax::MRD::MRD (  
    const char * serialnumber,  
    char ** outmessage = NULL ) [inline], [protected]
```

Base constructor.

## Parameters

<i>serialnumber</i>	The serial number of the device to open.
<i>outmessage</i>	To hold an error message, if any.

### 9.87.3.2 ~MRD()

```
azatrax::MRD::~~MRD ( ) [inline]
```

Base destructor.

## 9.87.4 Member Function Documentation

### 9.87.4.1 AllowingExternalChanges()

```
bool azatrax::MRD::AllowingExternalChanges ( ) const [inline]
```

Allowing External Changes?

References azatrax::MRD::status2\_union::allowExternalChanges, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateDataPacket::status2, azatrax::MRD::status2\_union::theBits, and azatrax::MRD::status2\_union::theByte.

### 9.87.4.2 ClearExternallyChanged()

```
ErrorCode azatrax::MRD::ClearExternallyChanged ( ) const [inline]
```

Clear 'ExternallyChanged' status bit.

Clear 'ExternallyChanged' status bit (data packet byte 2).

References azatrax::Azatrax::cmd\_ClearExternallyChanged, and azatrax::Azatrax::sendByte().

### 9.87.4.3 DisableExternal()

```
ErrorCode azatrax::MRD::DisableExternal ( ) const [inline]
```

Disable external changes of turnout state.

Disable external changes of turnout state (-S only).

References azatrax::Azatrax::cmd\_DisableExternal, and azatrax::Azatrax::sendByte().

#### 9.87.4.4 EnableExternal()

```
ErrorCode azatrax::MRD::EnableExternal ( ) const [inline]
```

Enable external changes of turnout state.

Enable external changes of turnout state (-S only).

References azatrax::Azatrax::cmd\_EnableExternal, and azatrax::Azatrax::sendByte().

#### 9.87.4.5 ExternallyChanged()

```
bool azatrax::MRD::ExternallyChanged ( ) const [inline]
```

Externally changed?

References azatrax::MRD::status2\_union::externallyChanged, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::↔StateDataPacket::status2, azatrax::MRD::status2\_union::theBits, and azatrax::MRD::status2\_union::theByte.

#### 9.87.4.6 HasRelays()

```
bool azatrax::MRD::HasRelays ( ) const [inline]
```

Has Relays?

References azatrax::MRD::status1\_union::modtype, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateData↔Packet::status1, azatrax::MRD::status1\_union::theBits, and azatrax::MRD::status1\_union::theByte.

#### 9.87.4.7 Identify\_1\_2()

```
ErrorCode azatrax::MRD::Identify_1_2 ( ) const [inline]
```

Identify 1 and 2.

Identify 1 and 2 - Flashes sensor 1 and 2 LEDs.

References azatrax::Azatrax::cmd\_Identify\_1\_2, and azatrax::Azatrax::sendByte().

#### 9.87.4.8 Identify\_2()

```
ErrorCode azatrax::MRD::Identify_2 ( ) const [inline]
```

Identify 2.

Identify 2 - Flashes sensor 2's LED.

References azatrax::Azatrax::cmd\_Identify\_2, and azatrax::Azatrax::sendByte().

#### 9.87.4.9 Latch\_1()

```
bool azatrax::MRD::Latch_1 ( ) const [inline]
```

Latch one.

References azatrax::MRD::status1\_union::latch\_1, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateData↔Packet::status1, azatrax::MRD::status1\_union::theBits, and azatrax::MRD::status1\_union::theByte.

#### 9.87.4.10 Latch\_2()

```
bool azatrax::MRD::Latch_2 ( ) const [inline]
```

Latch two.

References azatrax::MRD::status1\_union::latch\_2, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateData↔Packet::status1, azatrax::MRD::status1\_union::theBits, and azatrax::MRD::status1\_union::theByte.

#### 9.87.4.11 OperatingMode()

```
OperatingMode_Type azatrax::MRD::OperatingMode ( ) const [inline]
```

Operating mode.

Returns the operating mode.

References azatrax::Azatrax::StateDataPacket::operatingMode, and azatrax::Azatrax::stateDataPacket.

#### 9.87.4.12 ResetStatus()

```
bool azatrax::MRD::ResetStatus ( ) const [inline]
```

Reset status?

References azatrax::MRD::status2\_union::resetStatus, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateDataPacket::status2, azatrax::MRD::status2\_union::theBits, and azatrax::MRD::status2\_union::theByte.

#### 9.87.4.13 ResetStopwatch()

```
ErrorCode azatrax::MRD::ResetStopwatch ( ) const [inline]
```

Reset Stopwatch.

ResetStopwatch - Stops the stopwatch and resets time to 0.

References azatrax::Azatrax::cmd\_ResetStopwatch, and azatrax::Azatrax::sendByte().

#### 9.87.4.14 Sense\_1()

```
bool azatrax::MRD::Sense_1 ( ) const [inline]
```

Sensor one active.

References azatrax::MRD::status1\_union::sense\_1, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateDataPacket::status1, azatrax::MRD::status1\_union::theBits, and azatrax::MRD::status1\_union::theByte.

#### 9.87.4.15 Sense\_2()

```
bool azatrax::MRD::Sense_2 ( ) const [inline]
```

Sensor two active.

References azatrax::MRD::status1\_union::sense\_2, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateDataPacket::status1, azatrax::MRD::status1\_union::theBits, and azatrax::MRD::status1\_union::theByte.

#### 9.87.4.16 SetChan1()

```
ErrorCode azatrax::MRD::SetChan1 ( ) const [inline]
```

Set channel 1 relays and status bits.

Sets the relays and status bits as it a train activated channel 1 (-S, turnout mode only).

References azatrax::Azatrax::cmd\_SetChan1, and azatrax::Azatrax::sendByte().

#### 9.87.4.17 SetChan2()

```
ErrorCode azatrax::MRD::SetChan2 ( ) const [inline]
```

Set channel 2 relays and status bits.

Sets the relays and status bits as it a train activated channel 2 (-S, turnout mode only).

References azatrax::Azatrax::cmd\_SetChan2, and azatrax::Azatrax::sendByte().

#### 9.87.4.18 Stopwatch()

```
void azatrax::MRD::Stopwatch (
    uint8_t & fract,
    uint8_t & seconds,
    uint8_t & minutes,
    uint8_t & hours ) const [inline]
```

Stopwatch time.

Returns the current Stopwatch time.

##### Parameters

<i>fract</i>	1/100s of a second.
<i>seconds</i>	Whole seconds.
<i>minutes</i>	Whole minutes.
<i>hours</i>	Whole hours.

References azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateDataPacket::stopwatchHours, and azatrax::Azatrax::StateDataPacket::stopwatchMinutes.

## 9.87.4.19 StopwatchTicking()

```
bool azatrax::MRD::StopwatchTicking ( ) const [inline]
```

Stopwatch Ticking?

References azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateDataPacket::status2, azatrax::MRD::status2\_↵ union::stopwatchTicking, azatrax::MRD::status2\_union::theBits, and azatrax::MRD::status2\_union::theByte.

## 9.87.5 Friends And Related Function Documentation

## 9.87.5.1 Azatrax

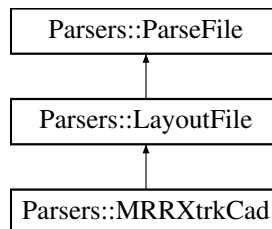
```
friend class Azatrax [friend]
```

## 9.88 Parsers::MRRXtrkCad Class Reference

[MRRXtrkCad](#) parser class.

```
#include <ParserClassesGroup.h>
```

Inheritance diagram for Parsers::MRRXtrkCad:



## Public Types

- enum [YY\\_MRRXtrkCad\\_ENUM\\_TOKEN](#) {  
[YY\\_MRRXtrkCad\\_NULL\\_TOKEN](#) =0, [INTEGER](#) =258, [FLOAT](#) =259, [STRING](#) =260,  
[RESTOFLINE](#) =261, [MULTILINE](#) =262, [EOL](#) =263, [UNTERMSTRING](#) =264,  
[NOTWORD](#) =265, [END](#) =266, [\\_VERSION](#) =267, [TITLE](#) =268,  
[MAPSCALE](#) =269, [ROOMSIZE](#) =270, [SCALE](#) =271, [HO](#) =272,  
[N](#) =273, [O](#) =274, [LAYERS](#) =275, [CURRENT](#) =276,  
[STRUCTURE](#) =277, [DRAW](#) =278, [CURVE](#) =279, [TURNOUT](#) =280,  
[TURNABLE](#) =281, [STRAIGHT](#) =282, [CAR](#) =283, [JOINT](#) =284,  
[NOTE](#) =285, [TEXT](#) =286, [MAIN](#) =287, [B](#) =288,  
[J](#) =289, [D](#) =290, [L](#) =291, [M](#) =292,  
[F](#) =293, [T](#) =294, [E](#) =295, [G](#) =296,  
[A](#) =297, [P](#) =298, [S](#) =299, [C](#) =300,  
[X](#) =301, [Y](#) =302, [Q](#) =303, [BLOCK](#) =304,  
[TRK](#) =305, [SWITCHMOTOR](#) =306 }

## Public Member Functions

- int [yyvsparse](#) (void)
- virtual void [yyerror](#) (char \*msg)
- virtual int [yylex](#) ()
- [MRRXtrkCad](#) (const char \*filename)  
*The constructor function.*
- virtual [~MRRXtrkCad](#) ()

## Public Attributes

- [yy\\_MRRXtrkCad\\_stype](#) yylval
- [yytype](#) yylloc
- int [yynerrs](#)
- int [yychar](#)
- int [yydebug](#)

## Private Member Functions

- int [lookup\\_word](#) (const char \*word) const
- void [yyerror1](#) (const char \*message, const char \*s) const

## Private Attributes

- bool [scanEol](#)
- bool [scanToEND](#)
- int [fieldflag](#)
- double [CurrentScale](#)

## Additional Inherited Members

### 9.88.1 Detailed Description

[MRRXtrkCad](#) parser class.

Include [MRRXtrkCad.tab.h](#) to get this class (the docs are wrong).

#### Author

Robert Heller <[heller@deepsoft.com](mailto:heller@deepsoft.com)>

### 9.88.2 Member Enumeration Documentation

#### 9.88.2.1 YY\_MRRXtrkCad\_ENUM\_TOKEN

enum [Parsers::MRRXtrkCad::YY\\_MRRXtrkCad\\_ENUM\\_TOKEN](#)



## Enumerator

YY_MRRXtrkCad_NULL_TOKEN	
INTEGER	
FLOAT	
STRING	
RESTOFLINE	
MULTILINE	
EOL	
UNTERMSTRING	
NOTWORD	
END	
_VERSION	
TITLE	
MAPSCALE	
ROOMSIZE	
SCALE	
HO	
N	
O	
LAYERS	
CURRENT	
STRUCTURE	
DRAW	
CURVE	
TURNOUT	
TURNTABLE	
STRAIGHT	
CAR	
JOINT	
NOTE	
TEXT	
MAIN	
B	
J	
D	
L	
M	
F	
T	
E	
G	
A	
P	
S	
C	
X	
Y	
Q	

**Enumerator**

BLOCK	
TRK	
SWITCHMOTOR	

### 9.88.3 Constructor & Destructor Documentation

#### 9.88.3.1 MRRXtrkCad()

```
Parsers::MRRXtrkCad::MRRXtrkCad (
    const char * filename )
```

The constructor function.

The constructor is the only function that is directly called from user code. See [LayoutFile](#) for all other access methods.

#### 9.88.3.2 ~MRRXtrkCad()

```
virtual Parsers::MRRXtrkCad::~~MRRXtrkCad ( ) [inline], [virtual]
```

References CurrentScale, fieldflag, lookup\_word(), scanEol, scanToEND, and yyerror1().

### 9.88.4 Member Function Documentation

#### 9.88.4.1 lookup\_word()

```
int Parsers::MRRXtrkCad::lookup_word (
    const char * word ) const [private]
```

Referenced by ~MRRXtrkCad().

#### 9.88.4.2 yyerror()

```
virtual void Parsers::MRRXtrkCad::yyerror (
    char * msg ) [virtual]
```

#### 9.88.4.3 yyerror1()

```
void Parsers::MRRXtrkCad::yyerror1 (
    const char * message,
    const char * s ) const [private]
```

Referenced by ~MRRXtrkCad().

#### 9.88.4.4 yylex()

```
virtual int Parsers::MRRXtrkCad::yylex ( ) [virtual]
```

#### 9.88.4.5 yyparse()

```
int Parsers::MRRXtrkCad::yyparse (
    void )
```

### 9.88.5 Member Data Documentation

#### 9.88.5.1 CurrentScale

```
double Parsers::MRRXtrkCad::CurrentScale [private]
```

Referenced by ~MRRXtrkCad().

#### 9.88.5.2 fieldflag

```
int Parsers::MRRXtrkCad::fieldflag [private]
```

Referenced by ~MRRXtrkCad().

**9.88.5.3 scanEol**

```
bool Parsers::MRRXtrkCad::scanEol [private]
```

Referenced by `~MRRXtrkCad()`.

**9.88.5.4 scanToEND**

```
bool Parsers::MRRXtrkCad::scanToEND [private]
```

Referenced by `~MRRXtrkCad()`.

**9.88.5.5 yychar**

```
int Parsers::MRRXtrkCad::yychar
```

**9.88.5.6 yydebug**

```
int Parsers::MRRXtrkCad::yydebug
```

**9.88.5.7 yylloc**

```
yy_ltype Parsers::MRRXtrkCad::yylloc
```

**9.88.5.8 yyval**

```
yy_MRRXtrkCad_stype Parsers::MRRXtrkCad::yyval
```

**9.88.5.9 yynerrs**

```
int Parsers::MRRXtrkCad::yynerrs
```

## 9.89 Icc::MTIDetail Class Reference

MTI Header type, detailed version.

### Public Member Functions

- [MTIDetail](#) (name,...)  
*Constructor: create a [MTIDetail](#) object.*
- [getHeader](#) (CANp=1)  
*Get the 29-bit CAN header or 16-bit MTI.*
- [setHeader](#) (header)  
*Decode the 29-bit header.*

### Private Attributes

- [mtiheader](#)  
*the [MTIHeader](#) component.*

### Static Private Attributes

- static [SPECIAL\\_MASK](#)  
*The Special bit is bit 13.*
- static [STREAMDG\\_MASK](#)  
*The Stream or Datagram bit is bit 12.*
- static [PRIORITY\\_SHIFT](#)  
*The priority is bits 10-11 of the MTI\_CAN.*
- static [PRIORITY\\_MASK](#)  
*The priority is bits 10-11 of the MTI\_CAN.*
- static [TYPEWITHIN\\_SHIFT](#)  
*The type within priority field is bits 5-9 of the MTI\_CAN.*
- static [TYPEWITHIN\\_MASK](#)  
*The type within priority field is bits 5-9 of the MTI\_CAN.*
- static [SIMPLE\\_SHIFT](#)  
*The simple bit is bit 4 of the MTI\_CAN.*
- static [SIMPLE\\_MASK](#)  
*The simple bit is bit 4 of the MTI\_CAN.*
- static [ADDRESSP\\_SHIFT](#)  
*The address present bit is bit 3 of the MTI\_CAN.*
- static [ADDRESSP\\_MASK](#)  
*The address present bit is bit 3 of the MTI\_CAN.*
- static [EVENTP\\_SHIFT](#)  
*The event present bit is bit 2 of the MTI\_CAN.*
- static [EVENTP\\_MASK](#)  
*The event present bit is bit 2 of the MTI\_CAN.*

- static [MODIFIER\\_SHIFT](#)  
*The modifier is bits 0-1 of the MTI\_CAN.*
- static [MODIFIER\\_MASK](#)  
*The modifier is bits 0-1 of the MTI\_CAN.*
- static [DESTID\\_SHIFT](#)  
*The destid is bits 0-11 of the MTI\_CAN.*
- static [DESTID\\_MASK](#)  
*The destid is bits 0-11 of the MTI\_CAN.*

### 9.89.1 Detailed Description

MTI Header type, detailed version.

Creates a 29-bit CAN header, specific to OpenLCB. The header is generated and decoded "on the fly" from/to the supplied options:

- -srcid A 12 bit source id field. Delegated to the mtiheader component.

See also

[lcc::CANHeader](#) and [lcc::MTIHeader](#).

- -special A boolean flag indicating if this is a special frame. Default is no.
- -streamordatagram A boolean flag indicating if this is a stream or datagram frame. Default is false.
- -priority A 2-bit integer specifying the frame's priority. Default is 0.
- -typewithin A 5-bit integer specifying the type withing the priority. Default is 0.
- -simple A boolean flag indicating if the frame is a simple protocol frame. Default is no.
- -addressp A boolean flag indicating if an address is present. Default is no.
- -eventp A boolean flag indicating if an event is present. Default is no.
- -modifier The 2-bit modifier field. Default is 0.
- -destid A 12-bit Desitination alias. Only used for stream and datagram frames. Default is 0.
- -datagramcontent An enumerated type defining the datagram or stream content type. Default is {} (not a datagram or stream).

### 9.89.2 Constructor & Destructor Documentation

#### 9.89.2.1 MTIDetail()

```
lcc::MTIDetail::MTIDetail (
    name ,
    ... )
```

Constructor: create a [MTIDetail](#) object.

A 29-bit CAN Header specific to the OpenLCB is created, using details for a MTI frame.

## Parameters

<i>name</i>	The name of the instance.
...	Options: <ul style="list-style-type: none"> <li>• -srcid A 12 bit source id field.</li> <li>• -special A boolean flag indicating if this is a special frame.</li> <li>• -streamordatagram A boolean flag indicating if this is a stream or datagram frame.</li> <li>• -priority A 2-bit integer specifying the frame's priority.</li> <li>• -typewithin A 5-bit integer specifying the type within the</li> <li>• -simple A boolean flag indicating if the frame is a simple protocol frame.</li> <li>• -addressp A boolean flag indicating if an address is present.</li> <li>• -eventp A boolean flag indicating if an event is present.</li> <li>• -modifier The 2-bit modifier field.</li> <li>• -destid A 12-bit Destination alias. Only used for stream and datagram frames.</li> <li>• -datagramcontent An enumerated type defining the datagram or stream content type.</li> </ul>

## 9.89.3 Member Function Documentation

## 9.89.3.1 getHeader()

```
lcc::MTIDetail::getHeader (
    CANp = 1 )
```

Get the 29-bit CAN header or 16-bit MTI.

Most of the heavy lifting is handled in the mtiheader component.

See also

[lcc::CANHeader](#) and [lcc::MTIHeader](#).

## Parameters

<i>CANp</i>	Specify whether we want a 29-bit CAN header or a 16-bit MTI.
-------------	--

**Returns**

The 29-bit CAN header or 16-bit MTI.

**9.89.3.2 setHeader()**

```
lcc::MTIDetail::setHeader (
    header )
```

Decode the 29-bit header.

Most of the heavy lifting is handled in the mtiheader component.

**See also**

[lcc::CANHeader](#) and [lcc::MTIHeader](#).

**Parameters**

<i>header</i>	The 29-bit header.
---------------	--------------------

**9.89.4 Member Data Documentation****9.89.4.1 ADDRESSP\_MASK**

```
lcc::MTIDetail::ADDRESSP_MASK [static], [private]
```

The address present bit is bit 3 of the MTI\_CAN.

**9.89.4.2 ADDRESSP\_SHIFT**

```
lcc::MTIDetail::ADDRESSP_SHIFT [static], [private]
```

The address present bit is bit 3 of the MTI\_CAN.



#### 9.89.4.3 DESTID\_MASK

```
lcc::MTIDetail::DESTID_MASK [static], [private]
```

The destid is bits 0-11 of the MTI\_CAN.

#### 9.89.4.4 DESTID\_SHIFT

```
lcc::MTIDetail::DESTID_SHIFT [static], [private]
```

The destid is bits 0-11 of the MTI\_CAN.

#### 9.89.4.5 EVENTP\_MASK

```
lcc::MTIDetail::EVENTP_MASK [static], [private]
```

The event present bit is bit 2 of the MTI\_CAN.

#### 9.89.4.6 EVENTP\_SHIFT

```
lcc::MTIDetail::EVENTP_SHIFT [static], [private]
```

The event present bit is bit 2 of the MTI\_CAN.

#### 9.89.4.7 MODIFIER\_MASK

```
lcc::MTIDetail::MODIFIER_MASK [static], [private]
```

The modifier is bits 0-1 of the MTI\_CAN.

#### 9.89.4.8 MODIFIER\_SHIFT

```
lcc::MTIDetail::MODIFIER_SHIFT [static], [private]
```

The modifier is bits 0-1 of the MTI\_CAN.

#### 9.89.4.9 mtiheader

```
lcc::MTIDetail::mtiheader [private]
```

the [MTIHeader](#) component.

Contains a [MTIHeader](#) to perform heavy lifting.

#### 9.89.4.10 PRIORITY\_MASK

```
lcc::MTIDetail::PRIORITY_MASK [static], [private]
```

The priority is bits 10-11 of the MTI\_CAN.

#### 9.89.4.11 PRIORITY\_SHIFT

```
lcc::MTIDetail::PRIORITY_SHIFT [static], [private]
```

The priority is bits 10-11 of the MTI\_CAN.

#### 9.89.4.12 SIMPLE\_MASK

```
lcc::MTIDetail::SIMPLE_MASK [static], [private]
```

The simple bit is bit 4 of the MTI\_CAN.

#### 9.89.4.13 SIMPLE\_SHIFT

```
lcc::MTIDetail::SIMPLE_SHIFT [static], [private]
```

The simple bit is bit 4 of the MTI\_CAN.

#### 9.89.4.14 SPECIAL\_MASK

```
lcc::MTIDetail::SPECIAL_MASK [static], [private]
```

The Special bit is bit 13.

#### 9.89.4.15 STREAMDG\_MASK

```
lcc::MTIDetail::STREAMDG_MASK [static], [private]
```

The Stream or Datagram bit is bit 12.

#### 9.89.4.16 TYPEWITHIN\_MASK

```
lcc::MTIDetail::TYPEWITHIN_MASK [static], [private]
```

The type within priority field is bits 5-9 of the MTI\_CAN.

#### 9.89.4.17 TYPEWITHIN\_SHIFT

```
lcc::MTIDetail::TYPEWITHIN_SHIFT [static], [private]
```

The type within priority field is bits 5-9 of the MTI\_CAN.

## 9.90 lcc::MTIHeader Class Reference

MTI Header type.

### Public Member Functions

- [MTIHeader](#) (name,...)  
*Constructor: create a [MTIHeader](#) A 29-bit CAN Header specific to the OpenLCB is created.*
- [getHeader](#) ()  
*Get the 29-bit header.*
- [setHeader](#) (header)  
*Decode the 29-bit header.*

### Private Attributes

- [canheader](#)  
*The [CANHeader](#) component.*

## Static Private Attributes

- static [MTI\\_CAN\\_SHIFT](#)  
*Bits 0-11 of the variable field are the MTI\_CAN field.*
- static [MTI\\_CAN\\_MASK](#)  
*Bits 0-11 of the variable field are the MTI\_CAN field.*
- static [FRAMETYPE\\_SHIFT](#)  
*Bits 12-14 of the variable field are the frame type field.*
- static [FRAMETYPE\\_MASK](#)  
*Bits 12-14 of the variable field are the frame type field.*

### 9.90.1 Detailed Description

MTI Header type.

Creates a 29-bit CAN header, specific to OpenLCB. The header is generated and decoded "on the fly" from/to the supplied options:

- -srcid A 12 bit source id field. Delegated to the canheader component.

See also

[lcc::CANHeader](#).

- -mti The 12 bit CAN\_MTI field. Default is 0, type is a 12-bit integer.
- -frametype The three bit frame type field. Default is 0, type is a 3-bit integer.

### 9.90.2 Constructor & Destructor Documentation

#### 9.90.2.1 MTIHeader()

```
lcc::MTIHeader::MTIHeader (
    name ,
    ... )
```

Constructor: create a [MTIHeader](#) A 29-bit CAN Header specific to the OpenLCB is created.

#### Parameters

<i>name</i>	The name of the instance.
...	Options: <ul style="list-style-type: none"> <li>• -srcid A 12 bit source id field.</li> <li>• -mti The 12 bit CAN_MTI field.</li> <li>• -frametype The three bit frame type field.</li> </ul>

### 9.90.3 Member Function Documentation

#### 9.90.3.1 getHeader()

```
lcc::MTIHeader::getHeader ( )
```

Get the 29-bit header.

Most of the heavy lifting is handled in the canheader component.

See also

[lcc::CANHeader](#).

Returns

The 29-bit header.

#### 9.90.3.2 setHeader()

```
lcc::MTIHeader::setHeader (
    header )
```

Decode the 29-bit header.

Most of the heavy lifting is handled in the canheader component.

See also

[lcc::CANHeader](#).

Parameters

<i>header</i>	The 29-bit header.
---------------	--------------------

## 9.90.4 Member Data Documentation

### 9.90.4.1 canheader

```
lcc::MTIHeader::canheader [private]
```

The [CANHeader](#) component.

Handles the header at the CAN level.

### 9.90.4.2 FRAMETYPE\_MASK

```
lcc::MTIHeader::FRAMETYPE_MASK [static], [private]
```

Bits 12-14 of the variable field are the frame type field.

### 9.90.4.3 FRAMETYPE\_SHIFT

```
lcc::MTIHeader::FRAMETYPE_SHIFT [static], [private]
```

Bits 12-14 of the variable field are the frame type field.

### 9.90.4.4 MTI\_CAN\_MASK

```
lcc::MTIHeader::MTI_CAN_MASK [static], [private]
```

Bits 0-11 of the variable field are the MTI\_CAN field.

### 9.90.4.5 MTI\_CAN\_SHIFT

```
lcc::MTIHeader::MTI_CAN_SHIFT [static], [private]
```

Bits 0-11 of the variable field are the MTI\_CAN field.

## 9.91 nce::NCE Class Reference

Main [NCE](#) Cab Bus interface class.

### Public Member Functions

- [NCE](#) (name, port="/dev/ttyS0",...)  
*Constructor.*
- [~NCE](#) ()  
*The destructor restores the serial port's state and closes it.*
- [NOP](#) ()  
*NOP, dummy instruction.*
- [AssignLoco](#) (locoaddress, cabnumber)  
*Assign loco to cab.*
- [ReturnClock](#) ()  
*Returns the fast clock to the RS232 port in binary mode.*
- [StopClock](#) ()  
*Stops the scale time clock.*
- [StartClock](#) ()  
*Starts the scale time clock.*
- [SetClock](#) (hours, minutes)  
*Set the scale time clock.*
- [SetClockFormat](#) (format)  
*Set clock 12/24 hours.*
- [SetClockRatio](#) (ratio)  
*Set clock speed (ratio).*
- [DequeuePacket](#) (locoaddress)  
*Dequeue packet by loco addr.*
- [EnableMain](#) ()  
*Enable main trk, kill prog.*
- [ReturnAuxiliaryInputUnit](#) (cabnumber)  
*Returns status of Auxiliary Input Unit.*
- [DisableMain](#) ()  
*Kill main track, enable program track.*
- [Dummy](#) ()  
*Dummy instruction returns "!" followed by CR/LF.*
- [SetLocoSpeedMode](#) (locoaddress, mode)  
*Sets the speed mode of loco.*
- [WriteToRAM](#) (address,...)  
*Writes bytes to a command station RAM address.*
- [ReadFromRAM](#) (address)  
*Returns 16 bytes from a RAM address.*
- [WriteLCDLine3](#) (cabnumber, textline)  
*Sends a message to LCD line 3 of a cab.*
- [WriteLCDLine4](#) (cabnumber, textline)  
*Sends a message to LCD line 4 of a cab.*

- [WriteLCDRightLine2](#) (cabnumber, textline)  
*Sends a message to the right side of LCD line 2 of a cab.*
- [WriteRAWPacket](#) (sendtimes,...)  
*Reads a raw packet to put in TEMP\_Q.*
- [WriteOneByteToRAM](#) (address, byte)  
*Writes 1 byte to a command station RAM address.*
- [WriteTwoBytesToRAM](#) (address, byte1, byte2)  
*Writes 2 bytes to a command station RAM address.*
- [Write4BytesToRAM](#) (address, byte1, byte2, byte3, byte4)  
*Writes 4 bytes to a command station RAM address.*
- [Write8BytesToRAM](#) (address,...)  
*Writes 4 bytes to a command station RAM address.*
- [ReturnAuxiliaryInputUnitShortForm](#) (cabnumber)  
*Returns status of Auxiliary Input Unit, short form.*
- [ExecuteMacro](#) (macroNumber)  
*Executes a previously defined macro for route control.*
- [ReadOneByteFromRAM](#) (address)  
*Reads 1 byte from a command station RAM address.*
- [ProgramMode](#) ()  
*Enters Program track mode.*
- [NormalMode](#) ()  
*Returns from Program track mode.*
- [WriteCVInPagedMode](#) (address, data)  
*Writes a CV in paged mode.*
- [ReadCVInPagedMode](#) (address)  
*Reads a CV in paged mode.*
- [SetLocomotiveSpeedAndDirection](#) (locoaddress, ssm, dir, speed)  
*Set locomotive speed and direction.*
- [SetLocomotiveFunctionsGroup1](#) (locoaddress, f0, f1, f2, f3, f4)  
*Set locomotive functions, group 1.*
- [SetLocomotiveFunctionsGroup2](#) (locoaddress, f5, f6, f7, f8)  
*Set locomotive functions, group 2.*
- [SetLocomotiveFunctionsGroup3](#) (locoaddress, f9, f10, f11, f12)  
*Set locomotive functions, group 3.*
- [AddLocomotiveToMultiUnit](#) (locoaddress, mtr, samedirection)  
*Add locomotive to Multi-Unit.*
- [RemoveLocomotiveFromMultiUnit](#) (locoaddress, mtr)  
*Remove locomotive to Multi-Unit.*
- [AddLeadLocomotiveToMultiUnit](#) (locoaddress, mtr, samedirection)  
*Add lead locomotive to Multi-Unit.*
- [AddRearLocomotiveToMultiUnit](#) (locoaddress, mtr, samedirection)  
*Add rear locomotive to Multi-Unit.*
- [ChangeMomentumLevel](#) (locoaddress, newlevel)  
*Change momentum level for loco or consist.*
- [WriteRAWTrackPacket](#) (...)  
*Reads a raw packet to put in TRK\_Q.*
- [WriteRegister](#) (register, data)



*Writes a register.*

- [ReadRegister](#) (register)

*Read a register.*

- [WriteCVInDirectMode](#) (address, data)

*Writes a CV in direct mode.*

- [ReadCVInDirectMode](#) (address)

*Reads a CV in direct mode.*

- [SoftwareVersion](#) ()

*Read software version number.*

- [SoftReset](#) ()

*Soft reset of command station.*

- [HardReset](#) ()

*Hard reset of command station.*

- [MacroCommand](#) (address, macronumber)

*Assign address to macro.*

- [AccessoryDecoderOperation](#) (address, activateOutput)

*Accessory decoder operation request.*

- [SetSignalAspect](#) (address, aspectBits)

*Set signal aspect.*

- [OperatingModeProgrammingByteModeWrite](#) (locoaddress, cv, data)

*Operating mode programming byte mode write.*

- [OperatingModeAccessoryProgrammingByteModeWrite](#) (address, cv, data)

*Operating mode accessory programming byte mode write.*

- [SetCabBusAddressOfUSBBoard](#) (cabaddress)

*Set the cab bus address of the USB board.*

- [SetBinaryCommandEchoMode](#) (mode)

*Set binary command echo mode.*

## Public Attributes

- [ttyfd](#)

*Terminal file descriptor.*

## Static Public Attributes

- static [NumberOfBytesReturned](#)

*Array containing the number of bytes expected for each command.*

## Private Member Functions

- [\\_transmit](#) (themessage)  
*Transmit a message.*
- [\\_readevent](#) ()  
*Read event handler, toggle timeout flag.*
- [\\_timeoutevent](#) ()  
*Timeout event handler, toggle timeout flag.*
- [\\_readbyte](#) (thebytevar, timeout=5)  
*Read next available byte or return false.*
- [\\_readresponse](#) (bufferVar, expectnumberofbytes)  
*Read a response message.*
- [\\_sendMessageAndReturnResponse](#) (message)  
*Send a message and return a response.*
- [\\_explodechars](#) (text)  
*Explode text into ASCII character codes.*

## Private Attributes

- [\\_timeout](#)  
*Timeout or data available flag.*

### 9.91.1 Detailed Description

Main [NCE](#) Cab Bus interface class.

This class implements the interface logic to connect to the [NCE](#) Cab Bus.

#### Parameters

<i>port</i>	Name of the serial port connected to the <a href="#">NCE</a> Cab Bus.
...	Options: <ul style="list-style-type: none"><li>• -baud Data rate, readonly, defaults to 9600, can be one of 9600 or 19200.</li></ul>

#### Author

Robert Heller <heller@deepsoft.com>

### 9.91.2 Constructor & Destructor Documentation

### 9.91.2.1 NCE()

```
nce::NCE::NCE (
    name ,
    port  = "/dev/ttyS0",
    ... )
```

Constructor.

#### Parameters

<i>port</i>	Name of the serial port connected to the <a href="#">NCE</a> Cab Bus.
...	Options: <ul style="list-style-type: none"><li>• -baud Data rate, readonly, defaults to 9600, can be one of 9600 or 19200.</li></ul>

### 9.91.2.2 ~NCE()

```
nce::NCE::~~NCE ( )
```

The destructor restores the serial port's state and closes it.

## 9.91.3 Member Function Documentation

### 9.91.3.1 \_explodechars()

```
nce::NCE::_explodechars (
    text ) [private]
```

Explode text into ASCII character codes.

### 9.91.3.2 \_readbyte()

```
nce::NCE::_readbyte (
    thebytevar ,
    timeout  = 5 ) [private]
```

Read next available byte or return false.

**Parameters**

<i>thebytevar</i>	Name of a variable to receive the byte.
<i>timeout</i>	Timeout in seconds.

If there is a defined external read event handler, the timeout parameter is ignored and false is returned if there are no bytes available. The presumption is that the read is being called from event handler and that means that there is data available.

**9.91.3.3 \_readevent()**

```
nce::NCE::_readevent ( ) [private]
```

Read event handler, toggle timeout flag.

**9.91.3.4 \_readresponse()**

```
nce::NCE::_readresponse (
    bufferVar ,
    expectnumberofbytes ) [private]
```

Read a response message.

**9.91.3.5 \_sendMessageAndReturnResponse()**

```
nce::NCE::_sendMessageAndReturnResponse (
    message ) [private]
```

Send a message and return a response.

**9.91.3.6 \_timeoutevent()**

```
nce::NCE::_timeoutevent ( ) [private]
```

Timeout event handler, toggle timeout flag.

### 9.91.3.7 \_transmit()

```
nce::NCE::_transmit (
    themessage ) [private]
```

Transmit a message.

### 9.91.3.8 AccessoryDecoderOperation()

```
nce::NCE::AccessoryDecoderOperation (
    address ,
    activateOutput )
```

Accessory decoder operation request.

#### Parameters

<i>address</i>	Accessory address (not in DCC format).
<i>activateOutput</i>	Output on or off.

#### Returns

The response message.

### 9.91.3.9 AddLeadLocomotiveToMultiUnit()

```
nce::NCE::AddLeadLocomotiveToMultiUnit (
    locoaddress ,
    mtr ,
    samedirection )
```

Add lead locomotive to Multi-Unit.

#### Parameters

<i>locoaddress</i>	Locomotive address.
<i>mtr</i>	Multi-Unit address.
<i>samedirection</i>	The locomotive direction is the same as the consist direction.

#### Returns

The response message.

#### 9.91.3.10 AddLocomotiveToMultiUnit()

```
nce::NCE::AddLocomotiveToMultiUnit (
    locoaddress ,
    mtr ,
    samedirection )
```

Add locomotive to Multi-Unit.

##### Parameters

<i>locoaddress</i>	Locomotive address.
<i>mtr</i>	Multi-Unit address.
<i>samedirection</i>	The locomotive direction is the same as the consist direction.

##### Returns

The response message.

#### 9.91.3.11 AddRearLocomotiveToMultiUnit()

```
nce::NCE::AddRearLocomotiveToMultiUnit (
    locoaddress ,
    mtr ,
    samedirection )
```

Add rear locomotive to Multi-Unit.

##### Parameters

<i>locoaddress</i>	Locomotive address.
<i>mtr</i>	Multi-Unit address.
<i>samedirection</i>	The locomotive direction is the same as the consist direction.

##### Returns

The response message.

#### 9.91.3.12 AssignLoco()

```
nce::NCE::AssignLoco (
    locoaddress ,
    cabnumber )
```

Assign loco to cab.

From Bincmds.pdf: Loco address for this command is always 2 bytes. The first byte is zero in the case of a short address. If the address is long then bits 6,7 of first byte must be set to 1

##### Parameters

<i>locoaddress</i>	Loco address (0-9999).
<i>cabnumber</i>	Cab number (0-63)

##### Returns

The response message.

#### 9.91.3.13 ChangeMomentumLevel()

```
nce::NCE::ChangeMomentumLevel (
    locoaddress ,
    newlevel )
```

Change momentum level for loco or consist.

##### Parameters

<i>locoaddress</i>	Locomotive or consist address.
<i>newlevel</i>	New momentum level (0-9).

##### Returns

The response message.

#### 9.91.3.14 DequeuePacket()

```
nce::NCE::DequeuePacket (
    locoaddress )
```

Dequeue packet by loco addr.

Reads loco address from BIN\_BUFF, finds the corresponding entry in TRK\_Q and deletes the packet from the TRK\_Q.

**Parameters**

<i>locoaddress</i>	Loco address (0-9999).
--------------------	------------------------

**Returns**

The response message.

**9.91.3.15 DisableMain()**

```
nce::NCE::DisableMain ( )
```

Kill main track, enable program track.

**Returns**

The response message.

**9.91.3.16 Dummy()**

```
nce::NCE::Dummy ( )
```

Dummy instruction returns "!" followed by CR/LF.

**Returns**

The response message.

**9.91.3.17 EnableMain()**

```
nce::NCE::EnableMain ( )
```

Enable main trk, kill prog.

**Returns**

The response message.

**9.91.3.18 ExecuteMacro()**

```
nce::NCE::ExecuteMacro (
    macroNumber )
```

Executes a previously defined macro for route control.



**Parameters**

<i>macroNumber</i>	The macro number.
--------------------	-------------------

**Returns**

The response message.

**9.91.3.19 HardReset()**

```
nce::NCE::HardReset ( )
```

Hard reset of command station.

Clears all RAM and resets command station to original fac defaults. All stored information is destroyed Note: the baud rate will be set to 9600

**9.91.3.20 MacroCommand()**

```
nce::NCE::MacroCommand (
    address ,
    macronumber )
```

Assign address to macro.

**Parameters**

<i>address</i>	Accessory address (not in DCC format).
<i>macronumber</i>	<a href="#">NCE</a> macro number (0-255).

**Returns**

The response message.

**9.91.3.21 NOP()**

```
nce::NCE::NOP ( )
```

NOP, dummy instruction.

**Returns**

The response message.

### 9.91.3.22 NormalMode()

```
nce::NCE::NormalMode ( )
```

Returns from Program track mode.

Power is restored to mainline and removed from program track. The queues are reinitialized for normal operation.

#### Returns

The response message.

### 9.91.3.23 OperatingModeAccessoryProgrammingByteModeWrite()

```
nce::NCE::OperatingModeAccessoryProgrammingByteModeWrite (
    address ,
    cv ,
    data )
```

Operating mode accessory programming byte mode write.

#### Parameters

<i>address</i>	Accessory address.
<i>cv</i>	CV to set.
<i>data</i>	Data to set.

#### Returns

The response message.

### 9.91.3.24 OperatingModeProgrammingByteModeWrite()

```
nce::NCE::OperatingModeProgrammingByteModeWrite (
    locoaddress ,
    cv ,
    data )
```

Operating mode programming byte mode write.

#### Parameters

<i>locoaddress</i>	Locomotive address.
<i>cv</i>	CV to set.
<i>data</i>	Data to set.

**Returns**

The response message.

**9.91.3.25 ProgramMode()**

```
nce::NCE::ProgramMode ( )
```

Enters Program track mode.

Power is removed from mainline and applied to program track. The queues are formatted to send reset packets.

**Returns**

The response message.

**9.91.3.26 ReadCVInDirectMode()**

```
nce::NCE::ReadCVInDirectMode (
    address )
```

Reads a CV in direct mode.

**Parameters**

<i>address</i>	CV address.
----------------	-------------

**Returns**

The register value, -1 for unsupported or -2 for bad CV number.

**9.91.3.27 ReadCVInPagedMode()**

```
nce::NCE::ReadCVInPagedMode (
    address )
```

Reads a CV in paged mode.

**Parameters**

<i>address</i>	CV address.
----------------	-------------

**Returns**

The register value, -1 for unsupported or -2 for bad CV number.

**9.91.3.28 ReadFromRAM()**

```
nce::NCE::ReadFromRAM (
    address )
```

Returns 16 bytes from a RAM address.

**Parameters**

<i>address</i>	Address to start reading from.
----------------	--------------------------------

**Returns**

The response message (16 data bytes).

**9.91.3.29 ReadOneByteFromRAM()**

```
nce::NCE::ReadOneByteFromRAM (
    address )
```

Reads 1 byte from a command station RAM address.

**Parameters**

<i>address</i>	RAM address.
----------------	--------------

**Returns**

The data byte.

#### 9.91.3.30 ReadRegister()

```
nce::NCE::ReadRegister (
    register )
```

Read a register.

##### Parameters

<i>register</i>	The register to read from.
-----------------	----------------------------

##### Returns

The register value, -1 for unsupported or -2 for bad register number.

#### 9.91.3.31 RemoveLocomotiveFromMultiUnit()

```
nce::NCE::RemoveLocomotiveFromMultiUnit (
    locoaddress ,
    mtr )
```

Remove locomotive to Multi-Unit.

##### Parameters

<i>locoaddress</i>	Locomotive address.
<i>mtr</i>	Multi-Unit address (not used).

##### Returns

The response message.

#### 9.91.3.32 ReturnAuxiliaryInputUnit()

```
nce::NCE::ReturnAuxiliaryInputUnit (
    cabnumber )
```

Returns status of Auxiliary Input Unit.

Returns four bytes. The first 2 bytes are a bit map of the 14 AIU inputs. The last 2 bytes are a bit map of any changes since this command was last given. If the cab is greater than 63 it will be "forced" to 0. The first time this command is given for a cab after the command station is powered up or reset the change bytes will be 0x3fff.

**Parameters**

<i>cabnumber</i>	Cab number (0-63)
------------------	-------------------

**Returns**

A list of two values, the bit map of values and the changed bitmap of values or the list {-1 -1} if the operation is unsupported.

**9.91.3.33 ReturnAuxiliaryInputUnitShortForm()**

```
nce::NCE::ReturnAuxiliaryInputUnitShortForm (
    cabnumber )
```

Returns status of Auxiliary Input Unit, short form.

This is a short form of CMD 0x8A. It returns only the first 2 bytes of command 0x8A. The 2 bytes are a bit map of the 14 AIU inputs. If the cab is greater than 63 it will be "forced" to 0.

**Parameters**

<i>cabnumber</i>	Cab number (0-63)
------------------	-------------------

**Returns**

The bit map of values or -1 if the operation is unsupported.

**9.91.3.34 ReturnClock()**

```
nce::NCE::ReturnClock ( )
```

Returns the fast clock to the RS232 port in binary mode.

**Returns**

The response message (hours, minutes).

**9.91.3.35 SetBinaryCommandEchoMode()**

```
nce::NCE::SetBinaryCommandEchoMode (
    mode )
```

Set binary command echo mode.

**Parameters**

<i>mode</i>	Mode to set: 0 = no echo, 1 = echo 1st byte of command, or 2 = echo entire command.
-------------	---

**Returns**

The response message.

**9.91.3.36 SetCabBusAddressOfUSBBoard()**

```
nce::NCE::SetCabBusAddressOfUSBBoard (
    cabaddress )
```

Set the cab bus address of the USB board.

**Parameters**

<i>cabaddress</i>	Cab address.
-------------------	--------------

**Returns**

The response message.

**9.91.3.37 SetClock()**

```
nce::NCE::SetClock (
    hours ,
    minutes )
```

Set the scale time clock.

**Parameters**

<i>hours</i>	Hours (0-23).
<i>minutes</i>	Minutes (0-59).

**Returns**

The response message.

**9.91.3.38 SetClockFormat()**

```
nce::NCE::SetClockFormat (
    format )
```

Set clock 12/24 hours.

**Parameters**

<i>format</i>	Clock format flag: true for 24 hour format, false for 12 hour format.
---------------	---

**Returns**

The response message.

**9.91.3.39 SetClockRatio()**

```
nce::NCE::SetClockRatio (
    ratio )
```

Set clock speed (ratio).

**Parameters**

<i>ratio</i>	Scale clock ratio, 1-25.
--------------	--------------------------

**Returns**

The response message.

**9.91.3.40 SetLocomotiveFunctionsGroup1()**

```
nce::NCE::SetLocomotiveFunctionsGroup1 (
    locoaddress ,
    f0 ,
    f1 ,
    f2 ,
    f3 ,
    f4 )
```

Set locomotive functions, group 1.



**Parameters**

<i>address</i>	Locomotive address.
<i>f0</i>	Function 0.
<i>f1</i>	Function 1.
<i>f2</i>	Function 2.
<i>f3</i>	Function 3.
<i>f4</i>	Function 4.

**Returns**

The response message.

**9.91.3.41 SetLocomotiveFunctionsGroup2()**

```
nce::NCE::SetLocomotiveFunctionsGroup2 (
    locoaddress ,
    f5 ,
    f6 ,
    f7 ,
    f8 )
```

Set locomotive functions, group 2.

**Parameters**

<i>address</i>	Locomotive address.
<i>f5</i>	Function 5.
<i>f6</i>	Function 6.
<i>f7</i>	Function 7.
<i>f8</i>	Function 7.

**Returns**

The response message.

**9.91.3.42 SetLocomotiveFunctionsGroup3()**

```
nce::NCE::SetLocomotiveFunctionsGroup3 (
    locoaddress ,
    f9 ,
```

```

    f10 ,
    f11 ,
    f12 )

```

Set locomotive functions, group 3.

#### Parameters

<i>address</i>	Locomotive address.
<i>f9</i>	Function 9.
<i>f10</i>	Function 10.
<i>f11</i>	Function 11.
<i>f12</i>	Function 12.

#### Returns

The response message.

#### 9.91.3.43 SetLocomotiveSpeedAndDirection()

```

nce::NCE::SetLocomotiveSpeedAndDirection (
    locoaddress ,
    ssm ,
    dir ,
    speed )

```

Set locomotive speed and direction.

#### Parameters

<i>locoaddress</i>	Locomotive address.
<i>ssm</i>	Speed mode (either S28 or S128).
<i>dir</i>	Direction (either Forward or Reverse).
<i>speed</i>	Locomotive speed (0-28, 0-128, or 255 (means emergency stop)).

#### Returns

The response message.

#### 9.91.3.44 SetLocoSpeedMode()

```

nce::NCE::SetLocoSpeedMode (
    locoaddress ,
    mode )

```

Sets the speed mode of loco.

#### Parameters

<i>locoaddress</i>	Loco address (0-9999).
<i>mode</i>	Speed step mode, one of S14, S28, or S128.

#### Returns

The response message.

#### 9.91.3.45 SetSignalAspect()

```
nce::NCE::SetSignalAspect (
    address ,
    aspectBits )
```

Set signal aspect.

#### Parameters

<i>address</i>	Accessory address (not in DCC format).
<i>aspectBits</i>	Signal aspect bit mask.

#### Returns

The response message.

#### 9.91.3.46 SoftReset()

```
nce::NCE::SoftReset ( )
```

Soft reset of command station.

Sets command Station to power up condition.

#### 9.91.3.47 SoftwareVersion()

```
nce::NCE::SoftwareVersion ( )
```

Read software version number.

#### Returns

The software version as three bytes.

**9.91.3.48 StartClock()**

```
nce::NCE::StartClock ( )
```

Starts the scale time clock.

**Returns**

The response message.

**9.91.3.49 StopClock()**

```
nce::NCE::StopClock ( )
```

Stops the scale time clock.

**Returns**

The response message.

**9.91.3.50 Write4BytesToRAM()**

```
nce::NCE::Write4BytesToRAM (
    address ,
    byte1 ,
    byte2 ,
    byte3 ,
    byte4 )
```

Writes 4 bytes to a command station RAM address.

**Parameters**

<i>address</i>	RAM address.
<i>byte1</i>	First byte to write.
<i>byte2</i>	Second byte to write.

**Returns**

The response message.

#### 9.91.3.51 Write8BytesToRAM()

```
nce::NCE::Write8BytesToRAM (
    address ,
    ... )
```

Writes 4 bytes to a command station RAM address.

##### Parameters

<i>address</i>	RAM address.
<i>byte1</i>	First byte to write.
<i>byte2</i>	Second byte to write.

##### Returns

The response message.

#### 9.91.3.52 WriteCVInDirectMode()

```
nce::NCE::WriteCVInDirectMode (
    address ,
    data )
```

Writes a CV in direct mode.

##### Parameters

<i>address</i>	CV address.
<i>data</i>	Data to write.

##### Returns

The response message.

#### 9.91.3.53 WriteCVInPagedMode()

```
nce::NCE::WriteCVInPagedMode (
    address ,
    data )
```

Writes a CV in paged mode.

**Parameters**

<i>address</i>	CV address.
<i>data</i>	Data to write.

**Returns**

The response message.

**9.91.3.54 WriteLCDLine3()**

```
nce::NCE::WriteLCDLine3 (
    cabnumber ,
    textline )
```

Sends a message to LCD line 3 of a cab.

**Parameters**

<i>cabnumber</i>	Cab Number (0-63).
<i>textline</i>	A string of 16 printable characters.

**Returns**

The response message.

**9.91.3.55 WriteLCDLine4()**

```
nce::NCE::WriteLCDLine4 (
    cabnumber ,
    textline )
```

Sends a message to LCD line 4 of a cab.

**Parameters**

<i>cabnumber</i>	Cab Number (0-63).
<i>textline</i>	A string of 16 printable characters.

**Returns**

The response message.

**9.91.3.56 WriteLCDRightLine2()**

```
nce::NCE::WriteLCDRightLine2 (
    cabnumber ,
    textline )
```

Sends a message to the right side of LCD line 2 of a cab.

**Parameters**

<i>cabnumber</i>	Cab Number (0-63).
<i>textline</i>	A string of 8 printable characters.

**Returns**

The response message.

**9.91.3.57 WriteOneByteToRAM()**

```
nce::NCE::WriteOneByteToRAM (
    address ,
    byte )
```

Writes 1 byte to a command station RAM address.

**Parameters**

<i>address</i>	RAM address.
<i>byte</i>	Byte to write.

**Returns**

The response message.

#### 9.91.3.58 WriteRAWPacket()

```
nce::NCE::WriteRAWPacket (
    sendtimes ,
    ... )
```

Reads a raw packet to put in TEMP\_Q.

##### Parameters

<i>sendtimes</i>	Number of times to send packet, 0 means don't send it. 255 is the same as 254 (system limitation).
...	Packet bytes to send, 3 to 6 bytes.

##### Returns

The response message.

#### 9.91.3.59 WriteRAWTrackPacket()

```
nce::NCE::WriteRAWTrackPacket (
    ... )
```

Reads a raw packet to put in TRK\_Q.

##### Parameters

...	Packet bytes to send, 3 to 5 bytes.
-----	-------------------------------------

##### Returns

The response message.

#### 9.91.3.60 WriteRegister()

```
nce::NCE::WriteRegister (
    register ,
    data )
```

Writes a register.



**Parameters**

<i>register</i>	The register to write to.
<i>data</i>	The data to write.

**Returns**

The response message.

**9.91.3.61 WriteToRAM()**

```
nce::NCE::WriteToRAM (
    address ,
    ... )
```

Writes bytes to a command station RAM address.

**Parameters**

<i>address</i>	Address to start writing to.
<i>...</i>	Bytes to write (upto 16).

**Returns**

The response message.

**9.91.3.62 WriteTwoBytesToRAM()**

```
nce::NCE::WriteTwoBytesToRAM (
    address ,
    byte1 ,
    byte2 )
```

Writes 2 bytes to a command station RAM address.

**Parameters**

<i>address</i>	RAM address.
<i>byte1</i>	First byte to write.
<i>byte2</i>	Second byte to write.

**Returns**

The response message.

**9.91.4 Member Data Documentation****9.91.4.1 `_timeout`**

```
nce::NCE::_timeout [private]
```

Timeout or data available flag.

**9.91.4.2 `NumberOfBytesReturned`**

```
nce::NCE::NumberOfBytesReturned [static]
```

Array containing the number of bytes expected for each command.

**9.91.4.3 `ttyfd`**

```
nce::NCE::ttyfd
```

Terminal file descriptor.

**9.92 `lcc::nid_or_null` Class Reference**

Node ID regexp pattern or the empty string.

**Static Public Member Functions**

- static [validate](#) (value)  
*Validate a Node ID, but allow a null string.*

**9.92.1 Detailed Description**

Node ID regexp pattern or the empty string.

A Node Id is six bytes as pairs of hex digits separated by colons (:).

## 9.92.2 Member Function Documentation

### 9.92.2.1 validate()

```
static lcc::nid_or_null::validate (
    value ) [static]
```

Validate a Node ID, but allow a null string.

#### Parameters

<i>value</i>	The value to validate, can be null.
--------------	-------------------------------------

## 9.93 Parsers::TrackGraph::NodeValues Struct Reference

Uncompressed graph node values.

### Public Member Functions

- [NodeValues](#) (int *\_id*=-1, [NodeType](#) *\_type*=[Undefined](#), [TurnoutGraphic](#) \**\_tgr*=NULL, [TurnoutRoutelist](#) \**\_tpo*=NULL, float *\_length*=0.0, [IntegerList](#) \**\_tracklist*=NULL, int *\_turnoutnumber*=0, char \**\_name*=NULL, char \**\_sensescript*=NULL, char \**\_normalactions*=NULL, char \**\_reverseactions*=NULL, int *\_numheads*=0, [StringPairList](#) \**\_aspects*=NULL, float *\_origx*=0.0, float *\_origy*=0.0, float *\_angle*=0.0, char \**\_onscript*=NULL, char \**\_offscript*=NULL)  
*Default constructor.*
- void [Cleanup](#) ()  
*Cleanup member function.*

### Public Attributes

- int [id](#)  
*Node number.*
- [NodeType](#) [type](#)  
*Type of node.*
- [TurnoutGraphic](#) \* [tgr](#)  
*Turnout graphic (if type is turnout).*
- [TurnoutRoutelist](#) \* [tpo](#)  
*Turnout route list (if type is turnout).*
- float [length](#)  
*Length of track.*
- [IntegerList](#) \* [tracklist](#)

*Track segments in block.*

- int [turnoutnumber](#)

*Number of the turnout.*

- char \* [name](#)

*Name of block or switch motor.*

- char \* [sensescript](#)

*Sense Script (occupation / point).*

- char \* [normalactionscript](#)

*Normal action script.*

- char \* [reverseactionscript](#)

*Reverse action script.*

- int [numheads](#)

*Number of heads.*

- StringPairList \* [aspectlist](#)

*Aspect list (name, script pairs).*

- float [origx](#)

*X coordinate of the base of the signal.*

- float [origy](#)

*Y coordinate of the base of the signal.*

- char \* [onscript](#)

*On action script.*

- char \* [offscript](#)

*Off action script.*

- float [angle](#)

*Angle of the signal.*

### 9.93.1 Detailed Description

Uncompressed graph node values.

Author

Robert Heller <heller@deepsoft.com>

### 9.93.2 Constructor & Destructor Documentation

### 9.93.2.1 NodeValues()

```
Parsers::TrackGraph::NodeValues::NodeValues (
    int _id = -1,
    NodeType _type = Undefined,
    TurnoutGraphic * _tgr = NULL,
    TurnoutRoutelist * _tpo = NULL,
    float _length = 0.0,
    IntegerList * _tracklist = NULL,
    int _turnoutnumber = 0,
    char * _name = NULL,
    char * _sensescript = NULL,
    char * _normalactionsscript = NULL,
    char * _reverseactionsscript = NULL,
    int _numheads = 0,
    StringPairList * _aspects = NULL,
    float _origx = 0.0,
    float _origy = 0.0,
    float _angle = 0.0,
    char * _onscript = NULL,
    char * _offscript = NULL ) [inline]
```

Default constructor.

## 9.93.3 Member Function Documentation

### 9.93.3.1 Cleanup()

```
void Parsers::TrackGraph::NodeValues::Cleanup ( ) [inline]
```

Cleanup member function.

## 9.93.4 Member Data Documentation

### 9.93.4.1 angle

```
float Parsers::TrackGraph::NodeValues::angle
```

Angle of the signal.

#### 9.93.4.2 aspectlist

```
StringPairList* Parsers::TrackGraph::NodeValues::aspectlist
```

Aspect list (name, script pairs).

#### 9.93.4.3 id

```
int Parsers::TrackGraph::NodeValues::id
```

Node number.

#### 9.93.4.4 length

```
float Parsers::TrackGraph::NodeValues::length
```

Length of track.

#### 9.93.4.5 name

```
char* Parsers::TrackGraph::NodeValues::name
```

Name of block or switch motor.

#### 9.93.4.6 normalactionscript

```
char* Parsers::TrackGraph::NodeValues::normalactionscript
```

Normal action script.

#### 9.93.4.7 numheads

```
int Parsers::TrackGraph::NodeValues::numheads
```

Number of heads.

#### 9.93.4.8 offscript

```
char* Parsers::TrackGraph::NodeValues::offscript
```

Off action script.

#### 9.93.4.9 onscript

```
char* Parsers::TrackGraph::NodeValues::onscript
```

On action script.

#### 9.93.4.10 origx

```
float Parsers::TrackGraph::NodeValues::origx
```

X coordinate of the base of the signal.

#### 9.93.4.11 origy

```
float Parsers::TrackGraph::NodeValues::origy
```

Y coordinate of the base of the signal.

#### 9.93.4.12 reverseactionscript

```
char* Parsers::TrackGraph::NodeValues::reverseactionscript
```

Reverse action script.

#### 9.93.4.13 sensescript

```
char* Parsers::TrackGraph::NodeValues::sensescript
```

Sense Script (occupation / point).

**9.93.4.14 tgr**

`TurnoutGraphic*` `Parsers::TrackGraph::NodeValues::tgr`

Turnout graphic (if type is turnout).

**9.93.4.15 tpo**

`TurnoutRoutelist*` `Parsers::TrackGraph::NodeValues::tpo`

Turnout route list (if type is turnout).

**9.93.4.16 tracklist**

`IntegerList*` `Parsers::TrackGraph::NodeValues::tracklist`

Track segments in block.

**9.93.4.17 turnoutnumber**

`int` `Parsers::TrackGraph::NodeValues::turnoutnumber`

Number of the turnout.

**9.93.4.18 type**

`NodeType` `Parsers::TrackGraph::NodeValues::type`

Type of node.

## 9.94 TTSupport::Occupied Class Reference

This class records a train sitting on a storage track during a specified time frame.

```
#include <Station.h>
```



## Public Member Functions

- [Occupied](#) (string trainnum\_="", double from\_=0.0, double until\_=0.0, string trainnum2\_="")  
*Constructor: record a train occupying a storage track.*
- const char \* [TrainNum](#) () const  
*Return the train that arrives.*
- const char \* [TrainNum2](#) () const  
*Return the train that departs.*
- double [From](#) () const  
*Return the start time;.*
- double [Until](#) () const  
*Return the end time.*
- [Occupied](#) (const [Occupied](#) &other)  
*Copy constructor – create an instance from another [Occupied](#) instance.*
- [Occupied](#) & [operator=](#) (const [Occupied](#) &other)  
*Assignment operator.*
- ostream & [Write](#) (ostream &stream) const  
*Write ourselves to an output stream.*
- istream & [Read](#) (istream &stream)  
*Read ourselves from an input stream.*

## Private Attributes

- string [trainnum](#)  
*The train that arrived.*
- string [trainnum2](#)  
*The train that departs.*
- double [from](#)  
*The start time of the occupation.*
- double [until](#)  
*The end time of the occupation.*

## Friends

- class [TimeTableSystem](#)  
*We are best buddies with the [TimeTableSystem](#) class.*

### 9.94.1 Detailed Description

This class records a train sitting on a storage track during a specified time frame.

The train number (symbol) might change when the train leaves the storage track.

#### Author

Robert Heller <heller@deepsoft.com>

## 9.94.2 Constructor & Destructor Documentation

### 9.94.2.1 Occupied() [1/2]

```
TTSupport::Occupied::Occupied (
    string trainnum_ = "",
    double from_ = 0.0,
    double until_ = 0.0,
    string trainnum2_ = "" ) [inline]
```

Constructor: record a train occupying a storage track.

#### Parameters

<i>trainnum_</i>	The arriving train number (symbol).
<i>from_</i>	The arrival time.
<i>until_</i>	The departure time.
<i>trainnum2_</i>	The departing train number (symbol). If it is the empty string, the departing train has the same number (symbol) as the arriving train.

References from, trainnum, trainnum2, and until.

### 9.94.2.2 Occupied() [2/2]

```
TTSupport::Occupied::Occupied (
    const Occupied & other ) [inline]
```

Copy constructor – create an instance from another [Occupied](#) instance.

#### Parameters

<i>other</i>	The other instance.
--------------	---------------------

References from, trainnum, trainnum2, and until.

## 9.94.3 Member Function Documentation

#### 9.94.3.1 From()

```
double TTSupport::Occupied::From ( ) const [inline]
```

Return the start time;.

References from.

#### 9.94.3.2 operator=()

```
Occupied& TTSupport::Occupied::operator= (
    const Occupied & other ) [inline]
```

Assignment operator.

Assign an [Occupied](#) instance to another [Occupied](#) instance.

##### Parameters

<i>other</i>	The other instance.
--------------	---------------------

References from, trainnum, trainnum2, and until.

#### 9.94.3.3 Read()

```
istream& TTSupport::Occupied::Read (
    istream & stream )
```

Read ourselves from an input stream.

##### Parameters

<i>stream</i>	The stream to read from.
---------------	--------------------------

Referenced by [TTSupport::Station::LastStorageTrack\(\)](#), [TTSupport::TimeRange::operator=\(\)](#), and [TTSupport::StorageTrack::operator=\(\)](#).

#### 9.94.3.4 TrainNum()

```
const char* TTSupport::Occupied::TrainNum ( ) const [inline]
```

Return the train that arrives.

References trainnum.

#### 9.94.3.5 TrainNum2()

```
const char* TTSupport::Occupied::TrainNum2 ( ) const [inline]
```

Return the train that departs.

References trainnum2.

#### 9.94.3.6 Until()

```
double TTSupport::Occupied::Until ( ) const [inline]
```

Return the end time.

References until.

#### 9.94.3.7 Write()

```
ostream& TTSupport::Occupied::Write (
    ostream & stream ) const
```

Write ourselves to an output stream.

##### Parameters

<i>stream</i>	The stream to write to.
---------------	-------------------------

Referenced by TTSupport::Station::LastStorageTrack(), TTSupport::TimeRange::operator=(), and TTSupport::↔StorageTrack::operator=().

## 9.94.4 Friends And Related Function Documentation

#### 9.94.4.1 TimeTableSystem

```
friend class TimeTableSystem [friend]
```

We are best buddies with the [TimeTableSystem](#) class.

### 9.94.5 Member Data Documentation

#### 9.94.5.1 from

```
double TTSupport::Occupied::from [private]
```

The start time of the occupation.

Referenced by [TTSupport::TimeRange::ContainsTime\(\)](#), [From\(\)](#), [TTSupport::TimeRange::From\(\)](#), [Occupied\(\)](#), [operator=\(\)](#), [TTSupport::TimeRange::operator=\(\)](#), [TTSupport::TimeRange::operator==\(\)](#), [TTSupport::TimeRange::operator<\(\)](#), [TTSupport::TimeRange::operator>\(\)](#), [TTSupport::StorageTrack::SetName\(\)](#), and [TTSupport::TimeRange::TimeRange\(\)](#).

#### 9.94.5.2 trainnum

```
string TTSupport::Occupied::trainnum [private]
```

The train that arrived.

Referenced by [Occupied\(\)](#), [operator=\(\)](#), and [TrainNum\(\)](#).

#### 9.94.5.3 trainnum2

```
string TTSupport::Occupied::trainnum2 [private]
```

The train that departs.

Referenced by [Occupied\(\)](#), [operator=\(\)](#), and [TrainNum2\(\)](#).

#### 9.94.5.4 until

```
double TTSupport::Occupied::until [private]
```

The end time of the occupation.

Referenced by `Occupied()`, `operator=()`, and `Until()`.

## 9.95 Icc::OpenLCBMessage Class Reference

OpenLCB Message type.

### Public Member Functions

- [OpenLCBMessage](#) (name,...)  
*Construct a OpenLCB Message object.*
- [toString](#) ()  
*Return the object as a printable string.*

### Static Public Member Functions

- static [validate](#) (object)  
*Validate an [OpenLCBMessage](#).*

### Private Member Functions

- [\\_configuredata](#) (option, value)  
*Configure method for data.*
- [\\_cgetdata](#) (option)  
*Cget method for data.*

### 9.95.1 Detailed Description

OpenLCB Message type.

Options (fields):

- -mti The MTI Header bitfield.
- -sourcenid The source Node ID.
- -destnid The destination Node ID or null if this is not an addressed message.
- -eventid The Event ID or null if there is no Event ID associated with this message.
- -data Any additional data associated with this message.

## 9.95.2 Constructor & Destructor Documentation

### 9.95.2.1 OpenLCBMessage()

```
lcc::OpenLCBMessage::OpenLCBMessage (
    name ,
    ... )
```

Construct a OpenLCB Message object.

#### Parameters

<i>name</i>	The name of the object
...	Options (fields): <ul style="list-style-type: none"><li>• -mti The MTI Header bitfield.</li><li>• -sourcenid The source Node ID.</li><li>• -destnid The destination Node ID or null if this is not an addressed message.</li><li>• -eventid The Event ID or null if there is no Event ID associated with this message.</li><li>• -data Any additional data associated with this message.</li></ul>

## 9.95.3 Member Function Documentation

### 9.95.3.1 \_cgetdata()

```
lcc::OpenLCBMessage::_cgetdata (
    option ) [private]
```

Cget method for data.

#### Parameters

<i>option</i>	Always -data.
---------------	---------------

**Returns**

Data vector (a list of bytes).

**9.95.3.2    \_configuredata()**

```
lcc::OpenLCBMessage::_configuredata (
    option ,
    value ) [private]
```

Configure method for data.

**Parameters**

<i>option</i>	Always -data.
<i>value</i>	A list of bytes.

**9.95.3.3    toString()**

```
lcc::OpenLCBMessage::toString ( )
```

Return the object as a printable string.

**Returns**

A string representation of the object.

**9.95.3.4    validate()**

```
static lcc::OpenLCBMessage::validate (
    object ) [static]
```

Validate an [OpenLCBMessage](#).

**Parameters**

<i>object</i>	A possible <a href="#">OpenLCBMessage</a> object.
---------------	---



## 9.96 lcc::OpenLCBNode Class Reference

Connect to a OpenLCB interface.

### Public Member Functions

- [OpenLCBNode](#) (name,...)  
*Constructor: construct a [OpenLCBNode](#) object.*
- [SendInitComplete](#) ()  
*Send an initialization complete message.*
- [SendVerifyNodeID](#) (nid="")  
*Send a verify node id message.*
- [ProtocolSupportRequest](#) (nid)  
*Send a Protocol Support Request to the specified node.*
- [ProduceEvent](#) (eventid)  
*Send an event message.*
- [IdentifyConsumer](#) (eventid)  
*Send an identify consumer message.*
- [ConsumerIdentified](#) (eventid, validity)  
*Send a consumer identified message.*
- [ConsumerRangeIdentified](#) (eventidrange)  
*Send a consumer range identified message.*
- [IdentifyProducer](#) (eventid)  
*Send an identify producer message.*
- [ProducerIdentified](#) (eventid, validity)  
*Send a producer identified message.*
- [ProducerRangeIdentified](#) (eventidrange)  
*Send a producer range identified message.*
- [IdentifyEvents](#) (nid="")  
*Send an identify events message.*
- [LearnEvent](#) (eventid)  
*Send a learn event message.*
- [SendDatagram](#) (nid, data)  
*Send a datagram message to the specified Node ID.*
- [DatagramReceivedOK](#) (nid, flags=0)  
*Send a datagram received ok message.*
- [DatagramRejected](#) (nid, errorcode)  
*Send a datagram rejected message.*
- [SendSimpleNodeInfoRequest](#) (nid)  
*Send a simple node info request message.*
- [SendSupportedProtocolsRequest](#) (nid)  
*Send a Supported protocols request message.*
- [SendMySupportedProtocols](#) (nid)  
*Send my supported protocols message.*
- [ReturnMySupportedProtocols](#) ()  
*Return my protocol support bitvector (three bytes).*

- [SendMySimpleNodeInfo](#) (nid)  
*Send my simple node info message.*
- [ReturnMySimpleNodeInfo](#) ()  
*Return my Simple Node Info payload.*
- [SendMyNodeVerification](#) ()  
*Send my node verification message.*

### Private Member Functions

- [\\_messageHandler](#) (message)  
*Generic message handler.*

### Static Private Member Functions

- static [nidlist](#) (nid)  
*Break a Node ID string into a list of bytes.*
- static [\\_buildSelectTransportConstructorDialog](#) ()  
*Build a dialog box to select the transport constructor.*
- static [\\_CancelSelectTransport](#) ()  
*Bound to the *Cancel* button.*
- static [\\_SelectTransport](#) ()  
*Bound to the *Select* button.*
- static [selectTransportConstructor](#) (...)  
*Pop up a select transport constructor dialog box.*
- static [transportConstructors](#) ()  
*Return the list of known available transport constructors.*

### Private Attributes

- [transport](#)  
*The logical transport layer component.*
- [protocolsupport](#)  
*Protocol support: Simple Protocol subset and SimpleNodeInfo by default.*
- [simplenodeinfo](#)  
*Simple node info payload.*

### Static Private Attributes

- static [\\_transportConstructors](#)  
*Array of transport constructors.*
- static [selectTransportConstructorDialog](#)  
*Dialog box to select the transport constructor.*
- static [constructorCombo](#)  
*[LabelComboBox](#) to select a transport constructor.*

### 9.96.1 Detailed Description

Connect to a OpenLCB interface.

This class implements the high level interface to the OpenLCB network.

Options:

- `-transport` The transport layer constructor.
- `-eventhandler` This is a script prefix that is run for event processing messages. This is a command with the procedure signature of

```
proc eventhandler {command eventid {{validity {}}}} {...}
```

The command is one of `consumerrangeidentified`, `consumeridentified`, `producerrangeidentified`, `produceridentified`, `learnevents`, `report`, `identifyconsumer`, `identifyproducer`, or `identifyevents`.

- `-datagramhandler` This is a script prefix that is run for datagram messages. This is a command with the procedure signature of

```
proc datagramhandler {command sourcenid args} {...}
```

The command argument is one of `datagramreceivedok`, `datagramrejected`, or `datagramcontent`

- `-generalmessagehandler` This is a script prefix that is run for general messages. This is a command with the procedure signature of

```
proc generalmessagehandler {message} {...}
```

The message is an [lcc::OpenLCBMessage](#) object. The procedure should reference the `-mti` option of the message object to determine what sort message it is.

Additional options are passed to the transport layer constructor.

### 9.96.2 Constructor & Destructor Documentation

#### 9.96.2.1 OpenLCBNode()

```
lcc::OpenLCBNode::OpenLCBNode (
    name ,
    ... )
```

Constructor: construct a [OpenLCBNode](#) object.

Open a connection to the OpenLCB network.

## Parameters

<i>name</i>	The name of the <a href="#">OpenLCBNode</a> object.
...	<p>Options:</p> <ul style="list-style-type: none"> <li>• <b>-transport</b> The transport layer constructor. This option is required.</li> <li>• <b>-eventhandler</b> This is a script prefix that is run for event processing messages. This is a command with the procedure signature of <pre>proc eventhandler {command eventid {{validity {}}}} {...}</pre> <p>The command is one of <code>consumerrangeidentified</code>, <code>consumeridentified</code>, <code>producerrangeidentified</code>, <code>produceridentified</code>, <code>learnerevents</code>, <code>report</code>, <code>identifyconsumer</code>, <code>identifyproducer</code>, or <code>identifyevents</code>.</p> </li> <li>• <b>-datagramhandler</b> This is a script prefix that is run for datagram messages. This is a command with the procedure signature of <pre>proc datagramhandler {command sourcenid data} {...}</pre> <p>The command argument is one of <code>datagramreceivedok</code>, <code>datagramrejected</code>, or <code>datagramcontent</code></p> </li> <li>• <b>-generalmessagehandler</b> This is a script prefix that is run for general messages. This is a command with the procedure signature of <pre>proc generalmessagehandler {message} {...}</pre> <p>The message is an <a href="#">lcc::OpenLCBMessage</a> object. The procedure should reference the <code>-mti</code> option of the message object to determine what sort message it is.</p> </li> </ul>

Additional options are passed to the transport layer constructor.

### 9.96.3 Member Function Documentation

#### 9.96.3.1 `_buildSelectTransportConstructorDialog()`

```
static lcc::OpenLCBNode::_buildSelectTransportConstructorDialog ( ) [static], [private]
```

Build a dialog box to select the transport constructor.

#### Returns

A transport constructor selection dialog box.

### 9.96.3.2 \_CancelSelectTransport()

```
static lcc::OpenLCBNode::_CancelSelectTransport ( ) [static], [private]
```

Bound to the `Cancel` button.

Closes the transport constructor dialog box and return the null string.

#### Returns

The null string.

### 9.96.3.3 \_messageHandler()

```
lcc::OpenLCBNode::_messageHandler (
    message ) [private]
```

Generic message handler.

#### Parameters

<i>message</i>	The received OpenLCB message.
----------------	-------------------------------

### 9.96.3.4 \_SelectTransport()

```
static lcc::OpenLCBNode::_SelectTransport ( ) [static], [private]
```

Bound to the `Select` button.

Closes the transport constructor dialog box and returns the selected transport constructor.

#### Returns

The transport constructor name.

### 9.96.3.5 ConsumerIdentified()

```
lcc::OpenLCBNode::ConsumerIdentified (
    eventid ,
    validity )
```

Send a consumer identified message.

## Parameters

<i>eventid</i>	The event id to send.
<i>validity</i>	The validity of the eventid: one of the words valid, invalid, or unknown.

**9.96.3.6 ConsumerRangeIdentified()**

```
lcc::OpenLCBNode::ConsumerRangeIdentified (
    eventidrange )
```

Send a consumer range identified message.

## Parameters

<i>eventidrange</i>	The event id range identified.
---------------------	--------------------------------

**9.96.3.7 DatagramReceivedOK()**

```
lcc::OpenLCBNode::DatagramReceivedOK (
    nid ,
    flags = 0 )
```

Send a datagram received ok message.

## Parameters

<i>nid</i>	The Node ID to send the message to.
<i>flags</i>	The (optional) flags to send.

**9.96.3.8 DatagramRejected()**

```
lcc::OpenLCBNode::DatagramRejected (
    nid ,
    errorcode )
```

Send a datagram rejected message.

## Parameters

<i>nid</i>	The Node ID to send the message to.
<i>errorcode</i>	The error code to send.

## 9.96.3.9 IdentifyConsumer()

```
lcc::OpenLCBNode::IdentifyConsumer (
    eventid )
```

Send an identify consumer message.

## Parameters

<i>eventid</i>	The event id to send.
----------------	-----------------------

## 9.96.3.10 IdentifyEvents()

```
lcc::OpenLCBNode::IdentifyEvents (
    nid = "" )
```

Send an identify events message.

## Parameters

<i>nid</i>	The (optional) Node ID to send the message to. If omitted, a global verify node id message is sent.
------------	---

## 9.96.3.11 IdentifyProducer()

```
lcc::OpenLCBNode::IdentifyProducer (
    eventid )
```

Send an identify producer message.

## Parameters

<i>eventid</i>	The event id to send.
----------------	-----------------------

#### 9.96.3.12 LearnEvent()

```
lcc::OpenLCBNode::LearnEvent (
    eventid )
```

Send a learn event message.

##### Parameters

<i>eventid</i>	The event id to learn.
----------------	------------------------

#### 9.96.3.13 nidlist()

```
static lcc::OpenLCBNode::nidlist (
    nid ) [static], [private]
```

Break a Node ID string into a list of bytes.

##### Parameters

<i>nid</i>	The Node ID to split up.
------------	--------------------------

#### 9.96.3.14 ProduceEvent()

```
lcc::OpenLCBNode::ProduceEvent (
    eventid )
```

Send an event message.

##### Parameters

<i>eventid</i>	The event id to send.
----------------	-----------------------

#### 9.96.3.15 ProducerIdentified()

```
lcc::OpenLCBNode::ProducerIdentified (
```



```
eventid ,  
validity )
```

Send a producer identified message.

#### Parameters

<i>eventid</i>	The event id to send.
<i>validity</i>	The validity of the eventid: one of the words valid, invalid, or unknown.

#### 9.96.3.16 ProducerRangeIdentified()

```
lcc::OpenLCBNode::ProducerRangeIdentified (   
    eventidrange )
```

Send a producer range identified message.

#### Parameters

<i>eventidrange</i>	The event id range identified.
---------------------	--------------------------------

#### 9.96.3.17 ProtocolSupportRequest()

```
lcc::OpenLCBNode::ProtocolSupportRequest (   
    nid )
```

Send a Protocol Support Request to the specified node.

#### Parameters

<i>nid</i>	The Node ID to send the message to.
------------	-------------------------------------

#### 9.96.3.18 ReturnMySimpleNodeInfo()

```
lcc::OpenLCBNode::ReturnMySimpleNodeInfo ( )
```

Return my Simple Node Info payload.

**Returns**

A Simple Node Info payload.

**9.96.3.19 ReturnMySupportedProtocols()**

```
lcc::OpenLCBNode::ReturnMySupportedProtocols ( )
```

Return my protocol support bitvector (three bytes).

**Returns**

A protocol support bitvector (three bytes).

**9.96.3.20 selectTransportConstructor()**

```
static lcc::OpenLCBNode::selectTransportConstructor (
    ... ) [static], [private]
```

Pop up a select transport constructor dialog box.

**Parameters**

...	Options:
	<ul style="list-style-type: none"> <li>-parent The parent window for this dialog box.</li> </ul>

**Returns**

Either the null string or the transport constructor.

**9.96.3.21 SendDatagram()**

```
lcc::OpenLCBNode::SendDatagram (
    nid ,
    data )
```

Send a datagram message to the specified Node ID.

## Parameters

<i>nid</i>	The Node ID to send the message to.
<i>data</i>	The data to send (1 to 64 bytes).

## 9.96.3.22 SendInitComplete()

```
lcc::OpenLCBNode::SendInitComplete ( )
```

Send an initialization complete message.

## 9.96.3.23 SendMyNodeVerification()

```
lcc::OpenLCBNode::SendMyNodeVerification ( )
```

Send my node verification message.

## 9.96.3.24 SendMySimpleNodeInfo()

```
lcc::OpenLCBNode::SendMySimpleNodeInfo (
    nid )
```

Send my simple node info message.

## Parameters

<i>nid</i>	The Node ID to send the message to.
------------	-------------------------------------

## 9.96.3.25 SendMySupportedProtocols()

```
lcc::OpenLCBNode::SendMySupportedProtocols (
    nid )
```

Send my supported protocols message.

**Parameters**

<i>nid</i>	The Node ID to send the message to.
------------	-------------------------------------

**9.96.3.26 SendSimpleNodeInfoRequest()**

```
lcc::OpenLCBNode::SendSimpleNodeInfoRequest (
    nid )
```

Send a simple node info request message.

**Parameters**

<i>nid</i>	The Node ID to send the message to.
------------	-------------------------------------

**9.96.3.27 SendSupportedProtocolsRequest()**

```
lcc::OpenLCBNode::SendSupportedProtocolsRequest (
    nid )
```

Send a Supported protocols request message.

**Parameters**

<i>nid</i>	The Node ID to send the message to.
------------	-------------------------------------

**9.96.3.28 SendVerifyNodeID()**

```
lcc::OpenLCBNode::SendVerifyNodeID (
    nid = "" )
```

Send a verify node id message.

**Parameters**

<i>nid</i>	The (optional) Node ID to send the message to. If omitted, a global verify node id message is sent.
------------	---

### 9.96.3.29 transportConstructors()

```
static lcc::OpenLCBNode::transportConstructors ( ) [static], [private]
```

Return the list of known available transport constructors.

#### Returns

The list of known available transport constructors as a description name list.

## 9.96.4 Member Data Documentation

### 9.96.4.1 \_transportConstructors

```
lcc::OpenLCBNode::_transportConstructors [static], [private]
```

Array of transport constructors.

### 9.96.4.2 constructorCombo

```
lcc::OpenLCBNode::constructorCombo [static], [private]
```

[LabelComboBox](#) to select a transport constructor.

### 9.96.4.3 protocolsupport

```
lcc::OpenLCBNode::protocolsupport [private]
```

Protocol support: Simple Protocol subset and SimpleNodeInfo by default.

#### 9.96.4.4 selectTransportConstructorDialog

```
lcc::OpenLCBNode::selectTransportConstructorDialog [static], [private]
```

Dialog box to select the transport constructor.

#### 9.96.4.5 simplenodeinfo

```
lcc::OpenLCBNode::simplenodeinfo [private]
```

Simple node info payload.

#### 9.96.4.6 transport

```
lcc::OpenLCBNode::transport [private]
```

The logical transport layer component.

## 9.97 lcc::OpenLCBOverTcp Class Reference

Connect to a OpenLCB over Tcp/Ip.

### Public Member Functions

- [OpenLCBOverTcp](#) (name,...)  
*Constructor: Connect to a Tcp/Ip OpenLCB network.*
- [setMessageHandler](#) (handler)  
*Set the message handler.*
- [sendMessage](#) (...)  
*Send a message on the OpenLCB bus.*

### Static Public Member Functions

- static [buildPortnidandhostDialog](#) ()  
*Function to construct the Dialog to ask the user for a port, host, and Node ID.*
- static [\\_CancelOpenTransport](#) ()  
*Function bound to the Cancel button.*
- static [\\_OpenTransport](#) ()  
*Function bound to the Open button.*
- static [requiredOpts](#) ()  
*Return the default option list.*
- static [drawOptionsDialog](#) (...)  
*Pop up the Options Dialog box.*

## Static Public Attributes

- static [portnidandhostDialog](#)  
*Dialog to ask the user for a port, host, and Node ID.*
- static [portLSpin](#)  
*LabelSpinBox containing possible network ports.*
- static [hostLEntry](#)  
*LabelEntry containing the hostname.*
- static [nidLEntry](#)  
*LabelEntry containing the Node ID.*

## Private Attributes

- [mtidetail](#)  
*MTIDetail component.*
- [messagehandler](#)  
*Message handler.*
- [datagrambuffers](#)  
*Datagram buffers.*
- [messagebuffers](#)  
*General message buffers (for multi frame messages)*
- [sock](#)  
*The socket I/O channel.*
- [\\_timeout](#)  
*Timeout flag.*

## Static Private Attributes

- static [NIDPATTERN](#)  
*The regexp for breaking up the Node ID into bytes.*

### 9.97.1 Detailed Description

Connect to a OpenLCB over Tcp/Ip.

Options:

- -host The name of the host (or IP address) to connect to. The default is localhost.
- -port The Tcp/Ip port number to connect with. The default is 12000.
- -nid The Node ID that the computer will assume in the format of `hh:hh:hh:hh:hh:hh` which is a 48 bit number expressed as 6 pairs of hexadecimal digits separated by colons (:).
- -promiscuousmode Promiscuous mode flag. If true all messages are handled, whether they are addressed to this node or not.

## 9.97.2 Constructor & Destructor Documentation

### 9.97.2.1 OpenLCBOverTcp()

```
lcc::OpenLCBOverTcp::OpenLCBOverTcp (
    name ,
    ... )
```

Constructor: Connect to a Tcp/Ip OpenLCB network.

Create a connection to a Tcp/Ip network.

#### Parameters

<i>name</i>	The name of the instance.
...	<p>The options:</p> <ul style="list-style-type: none"> <li>• -host The name of the host (or IP address) to connect to. The default is localhost.</li> <li>• -port The Tcp/Ip port number to connect with. The default is 12000.</li> <li>• -nid The Node ID that the computer will assume in the format of <code>hh:hh:hh:hh:hh:hh</code> which is a 48 bit number expressed as 6 pairs of hexadecimal digits separated by colons (:).</li> <li>• -promiscuousmode Promiscuous mode flag. If true all messages are handled, whether they are addressed to this node or not.</li> </ul>

## 9.97.3 Member Function Documentation

### 9.97.3.1 \_CancelOpenTransport()

```
static lcc::OpenLCBOverTcp::_CancelOpenTransport ( ) [static]
```

Function bound to the `Cancel` button.

Closes the dialog box and returns the empty string.

#### Returns

The empty string.



### 9.97.3.2 \_OpenTransport()

```
static lcc::OpenLCBOverTcp::_OpenTransport ( ) [static]
```

Function bound to the `Open` button.

Closes the dialog box and returns the options needed to open the transport.

#### Returns

An option argument list with the `-nid` and `-port` options.

### 9.97.3.3 buildPortnidandhostDialog()

```
static lcc::OpenLCBOverTcp::buildPortnidandhostDialog ( ) [static]
```

Function to construct the Dialog to ask the user for a port, host, and Node ID.

#### Returns

The Dialog box object.

### 9.97.3.4 drawOptionsDialog()

```
static lcc::OpenLCBOverTcp::drawOptionsDialog (
    ... ) [static]
```

Pop up the Options Dialog box.

Pops up the Options Dialog box and collects the options needed to open the [OpenLCBOverTcp](#) object.

#### Parameters

...	<p>Options:</p> <ul style="list-style-type: none"><li>• <code>-parent</code> Set the parent for this dialog box.</li><li>• <code>-port</code> The default Tcp/Ip port number option.</li><li>• <code>-host</code> The default Tcp/Ip hostname option.</li><li>• <code>-nid</code> The default Node ID to use for the Node ID option.</li></ul>
-----	--

**Returns**

Either the null string or an options list.

**9.97.3.5 requiredOpts()**

```
static lcc::OpenLCBOverTcp::requiredOpts ( ) [static]
```

Return the default option list.

Returns the default options for the options dialog.

**Returns**

The option value list.

**9.97.3.6 sendMessage()**

```
lcc::OpenLCBOverTcp::sendMessage (
    ... )
```

Send a message on the OpenLCB bus.

**Parameters**

...	Message options. See <a href="#">OpenLCBMessage</a> for possible options.
-----	---

**9.97.3.7 setMessageHandler()**

```
lcc::OpenLCBOverTcp::setMessageHandler (
    handler )
```

Set the message handler.

Generally called from the upper level class to gain access to incoming messages asynchronously.

#### Parameters

<i>handler</i>	The new handler procedure.
----------------	----------------------------

#### Returns

The old handler or the empty string if there was no old handler.

### 9.97.4 Member Data Documentation

#### 9.97.4.1 `_timeout`

```
lcc::OpenLCBOverTcp::_timeout [private]
```

Timeout flag.

#### 9.97.4.2 `datagrambuffers`

```
lcc::OpenLCBOverTcp::datagrambuffers [private]
```

Datagram buffers.

#### 9.97.4.3 `hostLEntry`

```
lcc::OpenLCBOverTcp::hostLEntry [static]
```

LabelEntry containing the hostname.

#### 9.97.4.4 `messagebuffers`

```
lcc::OpenLCBOverTcp::messagebuffers [private]
```

General message buffers (for multi frame messages)

#### 9.97.4.5 messagehandler

```
lcc::OpenLCBOverTcp::messagehandler [private]
```

Message handler.

#### 9.97.4.6 mtidetail

```
lcc::OpenLCBOverTcp::mtidetail [private]
```

[MTIDetail](#) component.

This component is used to extract and pack fields from and to a MTI header at a MTI detail level.

#### 9.97.4.7 nidLEntry

```
lcc::OpenLCBOverTcp::nidLEntry [static]
```

LabelEntry containing the Node ID.

#### 9.97.4.8 NIDPATTERN

```
lcc::OpenLCBOverTcp::NIDPATTERN [static], [private]
```

The regexp for breaking up the Node ID into bytes.

#### 9.97.4.9 portLSpin

```
lcc::OpenLCBOverTcp::portLSpin [static]
```

[LabelSpinBox](#) containing possible network ports.

#### 9.97.4.10 portnidandhostDialog

```
lcc::OpenLCBOverTcp::portnidandhostDialog [static]
```

Dialog to ask the user for a port, host, and Node ID.

#### 9.97.4.11 sock

```
lcc::OpenLCBOverTcp::sock [private]
```

The socket I/O channel.

## 9.98 lcc::OpenLCBProtocols Class Reference

Supported LCC Protocol name type.

### Static Public Member Functions

- static [validate](#) (object)  
*Validate a protocol name.*
- static [InsertProtocolBit](#) (bits, protocol)  
*Insert a protocol bit.*
- static [GetProtocolNames](#) (report)  
*Return a list of protocol names.*
- static [ProtocolLabelString](#) (protocolname)  
*Return a protocol label string.*

### Static Public Attributes

- static [protocolstrings](#)  
*Protocol display strings.*
- static [bitstype](#)  
*Protocol payload type.*

### 9.98.1 Detailed Description

Supported LCC Protocol name type.

### 9.98.2 Member Function Documentation

#### 9.98.2.1 GetProtocolNames()

```
static lcc::OpenLCBProtocols::GetProtocolNames (  
    report ) [static]
```

Return a list of protocol names.

**Parameters**

<i>report</i>	Protocol bit vector.
---------------	----------------------

**Returns**

A list of protocol names.

**9.98.2.2 InsertProtocolBit()**

```
static lcc::OpenLCBProtocols::InsertProtocolBit (
    bits ,
    protocol ) [static]
```

Insert a protocol bit.

**Parameters**

<i>bits</i>	Protocol bit vector.
<i>protocol</i>	Protocol bit to insert.

**Returns**

An updated protocol bit vector.

**9.98.2.3 ProtocolLabelString()**

```
static lcc::OpenLCBProtocols::ProtocolLabelString (
    protocolname ) [static]
```

Return a protocol label string.

**Parameters**

<i>protocolname</i>	The protocol name.
---------------------	--------------------

**Returns**

A human readable protocol label string.

#### 9.98.2.4 validate()

```
static lcc::OpenLCBProtocols::validate (
    object ) [static]
```

Validate a protocol name.

##### Parameters

<i>object</i>	Protocol name to validate.
---------------	----------------------------

### 9.98.3 Member Data Documentation

#### 9.98.3.1 bitstype

```
lcc::OpenLCBProtocols::bitstype [static]
```

Protocol payload type.

#### 9.98.3.2 protocolstrings

```
lcc::OpenLCBProtocols::protocolstrings [static]
```

Protocol display strings.

## 9.99 OvalWidgets::OvalButton Class Reference

Oval button.

### Public Member Functions

- [OvalButton](#) (name, \_canvas,...)  
*Construct an oval button.*
- [invoke](#) ()  
*Method for invoking the button.*
- [~OvalButton](#) ()  
*Destructor, free up all resources.*

## Private Member Functions

- [\\_ConfigureText](#) (option, value)  
*Method to configure the text of the button.*

## Private Attributes

- [canvas](#)  
*Canvas the widget is on.*

### 9.99.1 Detailed Description

Oval button.

Works just like a normal button widget.

#### Parameters

<a href="#">_canvas</a>	The canvas to draw the widget on.
...	<p>Options:</p> <ul style="list-style-type: none"><li>• -x The X coordinate (default 0).</li><li>• -y The Y coordinate (default 0).</li><li>• -width The width of the button (default 200).</li><li>• -height The height of the button (default 40).</li><li>• -background The background color (default white).</li><li>• -foreground The foreground color (default black).</li><li>• -fontfamily The font family (default Courier).</li><li>• -rightsquare Should the right end be square (default no)?</li><li>• -leftsquare Should the left end be square (default no)?</li><li>• -text The text of the button (default {}). </li><li>• -command The command of the button (default {}). </li></ul>

#### Author

Robert Heller <heller@deepsoft.com>

### 9.99.2 Constructor & Destructor Documentation



### 9.99.2.1 OvalButton()

```
OvalWidgets::OvalButton::OvalButton (
    name ,
    _canvas ,
    ... )
```

Construct an oval button.

#### Parameters

<i>_canvas</i>	The canvas to draw the button on.
<i>...</i>	The option value list.

### 9.99.2.2 ~OvalButton()

```
OvalWidgets::OvalButton::~~OvalButton ( )
```

Destructor, free up all resources.

## 9.99.3 Member Function Documentation

### 9.99.3.1 \_ConfigureText()

```
OvalWidgets::OvalButton::_ConfigureText (
    option ,
    value ) [private]
```

Method to configure the text of the button.

#### Parameters

<i>option</i>	The name of the option to configure.
<i>value</i>	The value to configure it to.

### 9.99.3.2 invoke()

```
OvalWidgets::OvalButton::invoke ( )
```

Method for invoking the button.

## 9.99.4 Member Data Documentation

### 9.99.4.1 canvas

```
OvalWidgets::OvalButton::canvas [private]
```

Canvas the widget is on.

## 9.100 OvalWidgets::OvalRoundCornerRectangle Class Reference

Oval Round Corner Rectangle.

### Public Member Functions

- [OvalRoundCornerRectangle](#) (name, \_canvas,...)  
*Construct an oval round corner rectangle.*
- [~OvalRoundCornerRectangle](#) ()  
*Destructor free up all resources.*

### Private Attributes

- [canvas](#)  
*Canvas the widget is on.*

### 9.100.1 Detailed Description

Oval Round Corner Rectangle.

Just a rectangle with rounded corners.

#### Parameters

<code>_canvas</code>	The canvas to draw the widget on.
<code>...</code>	Options: <ul style="list-style-type: none"> <li>• <code>-x</code> The X coordinate (default 0).</li> <li>• <code>-y</code> The Y coordinate (default 0).</li> </ul>
	<ul style="list-style-type: none"> <li>• <code>-width</code> The width of the button (default 200).</li> <li>• <code>-height</code> The height of the button (default 40).</li> <li>• <code>-color</code> The color of the rectangle (default white).</li> </ul>

**Author**

Robert Heller <heller@deepsoft.com>

**9.100.2 Constructor & Destructor Documentation****9.100.2.1 OvalRoundCornerRectangle()**

```
OvalWidgets::OvalRoundCornerRectangle::OvalRoundCornerRectangle (
    name ,
    _canvas ,
    ... )
```

Construct an oval round corner rectangle.

**Parameters**

<code>_canvas</code>	The canvas to draw the oval round corner rectangle on.
<code>...</code>	The option value list.

**9.100.2.2 ~OvalRoundCornerRectangle()**

```
OvalWidgets::OvalRoundCornerRectangle::~~OvalRoundCornerRectangle ( )
```

Destructor free up all resources.

**9.100.3 Member Data Documentation****9.100.3.1 canvas**

```
OvalWidgets::OvalRoundCornerRectangle::canvas [private]
```

Canvas the widget is on.

**9.101 OvalWidgets::OvalScale Class Reference**

An oval scale widget, much like a standard Tk scale widget.

## Public Member Functions

- [OvalScale](#) (name, \_canvas,...)  
*Constructor initialize and build an Oval Scale.*
- [~OvalScale](#) ()  
*Destructor free up all resources.*
- [set](#) (value)  
*Method to set the value of the scale.*
- [get](#) ()  
*Method to get the value of the scale.*

## Private Member Functions

- [\\_ConfigureXY](#) (option, value)  
*Method to configure an x or y coordinate.*
- [\\_ConfigureWL](#) (option, value)  
*Method to configure a width or length option.*
- [\\_ConfigureText](#) (option, value)  
*Method to configure the text of the button.*
- [\\_MoveThumb](#) (mx, my)  
*Method bound to button1 motion move the thumb.*
- [\\_BaseRect](#) (mx, my)  
*Method bound to button 1 presses.*

## Private Attributes

- [canvas](#)  
*Canvas the widget is on.*
- [\\_value](#)  
*Value of the scale.*

### 9.101.1 Detailed Description

An oval scale widget, much like a standard Tk scale widget.

#### Parameters

<a href="#">_canvas</a>	The canvas to draw the widget on.
-------------------------	-----------------------------------

## Parameters

...	Options: <ul style="list-style-type: none"> <li>• -x The X coordinate (default 0).</li> <li>• -y The Y coordinate (default 0).</li> <li>• -width The width of the scale (default 40).</li> <li>• -length The length of the scale (default 100).</li> <li>• -background The background color (default white).</li> <li>• -foreground The background color (default black).</li> <li>• -orientation The orientation of the scrollbar, horizontal or vertical (readonly, default vertical).</li> <li>• -from Start value of the scale (readonly, default 0).</li> <li>• -to End value of the scale (readonly, default 100).</li> <li>• -digits Number of digits to display (readonly, default 2).</li> <li>• -text Scale label (default "").</li> <li>• -showvalue Flag to indicate if the value should be displayed (readonly, default yes).</li> <li>• -variable Variable name to hold the value (default {}).</li> <li>• -bigincrement Large increment value (readonly, default 0).</li> <li>• -command The command of the scrollbar (default {}).</li> </ul>
-----	---

## Author

Robert Heller <heller@deepsoft.com>

## 9.101.2 Constructor &amp; Destructor Documentation

## 9.101.2.1 OvalScale()

```
OvalWidgets::OvalScale::OvalScale (
    name ,
    _canvas ,
    ... )
```

Constructor initialize and build an Oval Scale.

## Parameters

<code>_canvas</code>	The canvas to draw the scrollbar on.
...	The option value list.

### 9.101.2.2 ~OvalScale()

```
OvalWidgets::OvalScale::~~OvalScale ( )
```

Destructor free up all resources.

## 9.101.3 Member Function Documentation

### 9.101.3.1 \_BaseRect()

```
OvalWidgets::OvalScale::_BaseRect (
    mx ,
    my ) [private]
```

Method bound to button 1 presses.

#### Parameters

<i>mx</i>	Mouse X coordinate.
<i>my</i>	Mouse Y coordinate.

### 9.101.3.2 \_ConfigureText()

```
OvalWidgets::OvalScale::_ConfigureText (
    option ,
    value ) [private]
```

Method to configure the text of the button.

#### Parameters

<i>option</i>	The name of the option to configure.
<i>value</i>	The value to configure it to.

### 9.101.3.3 \_ConfigureWL()

```
OvalWidgets::OvalScale::_ConfigureWL (  
    option ,  
    value ) [private]
```

Method to configure a width or length option.

#### Parameters

<i>option</i>	The name of the option to configure.
<i>value</i>	The value of the option.

### 9.101.3.4 \_ConfigureXY()

```
OvalWidgets::OvalScale::_ConfigureXY (  
    option ,  
    value ) [private]
```

Method to configure an x or y coordinate.

#### Parameters

<i>option</i>	The name of the option to configure.
<i>value</i>	The value of the option.

### 9.101.3.5 \_MoveThumb()

```
OvalWidgets::OvalScale::_MoveThumb (  
    mx ,  
    my ) [private]
```

Method bound to button1 motion move the thumb.

#### Parameters

<i>mx</i>	Mouse X coordinate.
<i>my</i>	Mouse Y coordinate.

#### 9.101.3.6 `get()`

```
OvalWidgets::OvalScale::get ( )
```

Method to get the value of the scale.

#### 9.101.3.7 `set()`

```
OvalWidgets::OvalScale::set (
    value )
```

Method to set the value of the scale.

##### Parameters

<i>value</i>	The value to set the scale to.
--------------	--------------------------------

### 9.101.4 Member Data Documentation

#### 9.101.4.1 `_value`

```
OvalWidgets::OvalScale::_value [private]
```

Value of the scale.

#### 9.101.4.2 `canvas`

```
OvalWidgets::OvalScale::canvas [private]
```

Canvas the widget is on.

## 9.102 `OvalWidgets::OvalSlider` Class Reference

Oval Slider.



## Public Member Functions

- [OvalSlider](#) (name, \_canvas,...)  
*Construct an oval button.*
- [set](#) (value)  
*Method to set the value of the slider.*
- [get](#) ()  
*Method to get the current value.*

## Private Member Functions

- [\\_ConfigureL](#) (option, value)  
*Method to configure the length option.*
- [\\_ConfigureText](#) (option, value)  
*Method to configure the text of the button.*
- [\\_VerifyBitmap](#) (option, value)  
*Method to validate a bitmap option.*
- [\\_MoveThumb](#) (mx, my)  
*Method bound to the button 1 motion on the thumb.*

## Private Attributes

- [canvas](#)  
*Canvas the widget is on.*
- [\\_value](#)  
*Value of the scale.*

### 9.102.1 Detailed Description

Oval Slider.

This is like the activation control for the Star Trek NG Transporter.

#### Parameters

<a href="#">_canvas</a>	The canvas to draw the widget on.
-------------------------	-----------------------------------

## Parameters

...	<p>Options:</p> <ul style="list-style-type: none"> <li>• -x The X coordinate (default 0).</li> <li>• -y The Y coordinate (default 0).</li> <li>• -width The width of the slider button (default 200).</li> <li>• -height The height of the slider button (default 40).</li> <li>• -length The length of the slider (default 600).</li> <li>• -background The background color (default white).</li> <li>• -foreground The background color (default black).</li> <li>• -fontfamily The font family (default Courier).</li> <li>• -rightsquare Should the right end be square (default no)?</li> <li>• -leftsquare Should the left end be square (default no)?</li> <li>• -text The text of the button (default {}).</li> <li>• -command The command of the button (default {}).</li> <li>• -from Start value of the scale (readonly, default 0).</li> <li>• -to End value of the scale (readonly, default 100).</li> <li>• -digits Number of digits to display (readonly, default 2).</li> <li>• -showvalue Flag to indicate if the value should be displayed (readonly, default yes).</li> <li>• -variable Variable name to hold the value (default {}).</li> <li>• -stipple Stipple bitmap to use (readonly, default HBar).</li> </ul>
-----	--

## Author

Robert Heller <heller@deepsoft.com>

## 9.102.2 Constructor &amp; Destructor Documentation

## 9.102.2.1 OvalSlider()

```
OvalWidgets::OvalSlider::OvalSlider (
    name ,
    _canvas ,
    ... )
```

Construct an oval button.

## Parameters

<i>_canvas</i>	The canvas to draw the button on.
<i>...</i>	The option value list.

### 9.102.3 Member Function Documentation

#### 9.102.3.1 \_ConfigureL()

```
OvalWidgets::OvalSlider::_ConfigureL (  
    option ,  
    value ) [private]
```

Method to configure the length option.

## Parameters

<i>option</i>	The name of the option to configure.
<i>value</i>	The value of the option.

#### 9.102.3.2 \_ConfigureText()

```
OvalWidgets::OvalSlider::_ConfigureText (  
    option ,  
    value ) [private]
```

Method to configure the text of the button.

## Parameters

<i>option</i>	The name of the option to configure.
<i>value</i>	The value to configure it to.

#### 9.102.3.3 \_MoveThumb()

```
OvalWidgets::OvalSlider::_MoveThumb (  

```

```
mx ,  
my ) [private]
```

Method bound to the button 1 motion on the thumb.

#### Parameters

<i>mx</i>	Mouse X value.
<i>my</i>	Mouse Y value.

#### 9.102.3.4 \_VerifyBitmap()

```
OvalWidgets::OvalSlider::_VerifyBitmap (  
    option ,  
    value ) [private]
```

Method to validate a bitmap option.

#### Parameters

<i>option</i>	The name of the option to validate.
<i>value</i>	The value to validate.

#### 9.102.3.5 get()

```
OvalWidgets::OvalSlider::get ( )
```

Method to get the current value.

#### 9.102.3.6 set()

```
OvalWidgets::OvalSlider::set (  
    value )
```

Method to set the value of the slider.

#### Parameters

<i>value</i>	The value to set.
--------------	-------------------

### 9.102.4 Member Data Documentation

#### 9.102.4.1 `_value`

OvalWidgets::OvalSlider::\_value [private]

Value of the scale.

#### 9.102.4.2 `canvas`

OvalWidgets::OvalSlider::canvas [private]

Canvas the widget is on.

## 9.103 OvalWidgets::OvalScrollBar Class Reference

Oval ScrollBar.

### Public Member Functions

- [OvalScrollBar](#) (name, \_canvas,...)  
*Constructor initialize and build an Oval Scrollbar.*
- [~OvalScrollBar](#) ()  
*Destructor free up all resources.*
- [resize](#) (newMin, newMax)  
*Resize method.*
- [delta](#) (deltaX, deltaY)  
*Method to return a real number indicating the fractional change in the scrollbar setting that corresponds to a given change in slider position.*
- [fraction](#) (x, y)  
*Method to return a real number between 0 and 1 indicating where the point given by x and y lies in the trough area of the scrollbar.*
- [get](#) ()  
*Method to return the scrollbar settings in the form of a list whose elements are the arguments to the most recent set widget command.*
- [identify](#) (x, y)  
*Method to return the name of the element under the point given by x and y (such as arrow1), or an empty string if the point does not lie in any element of the scrollbar.*
- [set](#) (first, last)  
*This method is invoked by the scrollbar's associated widget to tell the scrollbar about the current view in the widget.*

## Private Member Functions

- [\\_ConfigureXY](#) (option, value)  
*Method to configure an x or y coordinate.*
- [\\_ConfigureWL](#) (option, value)  
*Method to configure a width or length option.*
- [\\_MoveThumb](#) (mx, my)  
*Method bound to button1 motion move the thumb.*
- [\\_BaseRect](#) (mx, my)  
*Method bound to button 1 presses.*
- [\\_Command](#) (...)  
*Method used to invoke the command as the thumb is moved.*

## Private Attributes

- [canvas](#)  
*Canvas the widget is on.*
- [\\_lastSet](#)  
*Holds the last scrollbar settings.*

### 9.103.1 Detailed Description

Oval ScrollBar.

Works just like a normal scrollbar widget.

#### Parameters

<a href="#">_canvas</a>	The canvas to draw the widget on.
...	Options: <ul style="list-style-type: none"> <li>• -x The X coordinate (default 0).</li> <li>• -y The Y coordinate (default 0).</li> <li>• -width The width of the scrollbar (default 40).</li> <li>• -length The length of the scrollbar (default 100).</li> <li>• -background The background color (default white).</li> <li>• -foreground The background color (default black).</li> <li>• -orientation The orientation of the scrollbar, horizontal or vertical (readonly, default vertical).</li> <li>• -command The command of the scrollbar (default {}). </li> </ul>

**Author**

Robert Heller <heller@deepsoft.com>

**9.103.2 Constructor & Destructor Documentation****9.103.2.1 OvalScrollBar()**

```
OvalWidgets::OvalScrollBar::OvalScrollBar (
    name ,
    _canvas ,
    ... )
```

Constructor initialize and build an Oval Scrollbar.

**Parameters**

<code>_canvas</code>	The canvas to draw the scrollbar on.
<code>...</code>	The option value list.

**9.103.2.2 ~OvalScrollBar()**

```
OvalWidgets::OvalScrollBar::~~OvalScrollBar ( )
```

Destructor free up all resources.

**9.103.3 Member Function Documentation****9.103.3.1 \_BaseRect()**

```
OvalWidgets::OvalScrollBar::_BaseRect (
    mx ,
    my ) [private]
```

Method bound to button 1 presses.

**Parameters**

<i>mx</i>	Mouse X coordinate.
<i>my</i>	Mouse Y coordinate.

**9.103.3.2 \_Command()**

```
OvalWidgets::OvalScrollBar::_Command (  
    ... ) [private]
```

Method used to invoke the command as the thumb is moved.

**Parameters**

...	passed to -command option.
-----	----------------------------

**9.103.3.3 \_ConfigureWL()**

```
OvalWidgets::OvalScrollBar::_ConfigureWL (  
    option ,  
    value ) [private]
```

Method to configure a width or length option.

**Parameters**

<i>option</i>	The name of the option to configure.
<i>value</i>	The value of the option.

**9.103.3.4 \_ConfigureXY()**

```
OvalWidgets::OvalScrollBar::_ConfigureXY (  
    option ,  
    value ) [private]
```

Method to configure an x or y coordinate.



## Parameters

<i>option</i>	The name of the option to configure.
<i>value</i>	The value of the option.

9.103.3.5 `_MoveThumb()`

```
OvalWidgets::OvalScrollBar::_MoveThumb (
    mx ,
    my ) [private]
```

Method bound to button1 motion move the thumb.

## Parameters

<i>mx</i>	Mouse X coordinate.
<i>my</i>	Mouse Y coordinate.

9.103.3.6 `delta()`

```
OvalWidgets::OvalScrollBar::delta (
    deltaX ,
    deltaY )
```

Method to return a real number indicating the fractional change in the scrollbar setting that corresponds to a given change in slider position.

For example, if the scrollbar is horizontal, the result indicates how much the scrollbar setting must change to move the slider `deltaX` pixels to the right (`deltaY` is ignored in this case). If the scrollbar is vertical, the result indicates how much the scrollbar setting must change to move the slider `deltaY` pixels down. The arguments and the result may be zero or negative.

## Parameters

<i>deltaX</i>	Amount of movement if scrollbar is horizontal.
<i>deltaY</i>	Amount of movement if scrollbar is vertical.

9.103.3.7 `fraction()`

```
OvalWidgets::OvalScrollBar::fraction (
```

```

    x ,
    y )

```

Method to return a real number between 0 and 1 indicating where the point given by x and y lies in the trough area of the scrollbar.

The value 0 corresponds to the top or left of the trough, the value 1 corresponds to the bottom or right, 0.5 corresponds to the middle, and so on. X and y must be pixel coordinates relative to the scrollbar widget. If x and y refer to a point outside the trough, the closest point in the trough is used.

#### Parameters

x	The X coordinate to check.
y	The Y coordinate to check.

#### 9.103.3.8 get()

```
OvalWidgets::OvalScrollBar::get ( )
```

Method to return the scrollbar settings in the form of a list whose elements are the arguments to the most recent set widget command.

#### 9.103.3.9 identify()

```

OvalWidgets::OvalScrollBar::identify (
    x ,
    y )

```

Method to return the name of the element under the point given by x and y (such as arrow1), or an empty string if the point does not lie in any element of the scrollbar.

X and y must be pixel coordinates relative to the scrollbar widget.

#### Parameters

x	The X coordinate to check.
y	The Y coordinate to check.

#### 9.103.3.10 resize()

```
OvalWidgets::OvalScrollBar::resize (
```

```
newMin ,  
newMax )
```

Resize method.

Method update the range of the scroll region.

#### Parameters

<i>newMin</i>	new minimum of the scroll region.
<i>newMax</i>	new maximum of the scroll region.

#### 9.103.3.11 set()

```
OvalWidgets::OvalScrollBar::set (  
    first ,  
    last )
```

This method is invoked by the scrollbar's associated widget to tell the scrollbar about the current view in the widget.

The command takes two arguments, each of which is a real fraction between 0 and 1. The fractions describe the range of the document that is visible in the associated widget. For example, if first is 0.2 and last is 0.4, it means that the first part of the document visible in the window is 20% of the way through the document, and the last visible part is 40% of the way through.

#### Parameters

<i>first</i>	First visible fraction.
<i>last</i>	Last visible fraction.

### 9.103.4 Member Data Documentation

#### 9.103.4.1 \_lastSet

```
OvalWidgets::OvalScrollBar::_lastSet [private]
```

Holds the last scrollbar settings.

#### 9.103.4.2 canvas

```
OvalWidgets::OvalScrollBar::canvas [private]
```

Canvas the widget is on.

## 9.104 FCFSupport::Owner Class Reference

The [Owner](#) class describes a car owner.

```
#include <Owner.h>
```

### Public Member Functions

- [Owner](#) ()  
*The default constructor initializes all fields to the empty string.*
- [Owner](#) ([Owner](#) &other)  
*The copy constructor copies the contents of another [Owner](#) to this one.*
- [Owner](#) & [operator=](#) ([Owner](#) &other)  
*The Assignment operator copies the contents of another [Owner](#) to this one.*
- [Owner](#) (const char \*i, const char \*n, const char \*c)  
*The full constructor initializes the class instance from user supplied parameters.*
- [~Owner](#) ()  
*The destructor does nothing special.*
- const char \* [Initials](#) () const  
*Return this owner's initials.*
- const char \* [Name](#) () const  
*Return this owner's name.*
- const char \* [Comment](#) () const  
*Return commentary about this owner.*

### Private Attributes

- string [initials](#)  
*This owner's initials.*
- string [name](#)  
*This owner's name.*
- string [comment](#)  
*Commentary about this owner.*

### 9.104.1 Detailed Description

The [Owner](#) class describes a car owner.

A car owned has a set of (three letter) initials, a full name, and an (optional) comment. This information is just used for informational purposes.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.104.2 Constructor & Destructor Documentation

#### 9.104.2.1 Owner() [1/3]

```
FCFSupport::Owner::Owner ( ) [inline]
```

The default constructor initializes all fields to the empty string.

References comment, initials, and name.

#### 9.104.2.2 Owner() [2/3]

```
FCFSupport::Owner::Owner (
    Owner & other ) [inline]
```

The copy constructor copies the contents of another [Owner](#) to this one.

#### Parameters

<i>other</i>	The other <a href="#">Owner</a> object.
--------------	---

References comment, initials, and name.

#### 9.104.2.3 Owner() [3/3]

```
FCFSupport::Owner::Owner (
    const char * i,
```

```
const char * n,
const char * c ) [inline]
```

The full constructor initializes the class instance from user supplied parameters.

#### Parameters

<i>i</i>	The owner's initials.
<i>n</i>	The owner's name.
<i>c</i>	Commentary about this owner.

References comment, i, initials, and name.

#### 9.104.2.4 ~Owner()

```
FCFSupport::Owner::~~Owner ( ) [inline]
```

The destructor does nothing special.

### 9.104.3 Member Function Documentation

#### 9.104.3.1 Comment()

```
const char* FCFSupport::Owner::Comment ( ) const [inline]
```

Return commentary about this owner.

References comment.

#### 9.104.3.2 Initials()

```
const char* FCFSupport::Owner::Initials ( ) const [inline]
```

Return this owner's initials.

References initials.

### 9.104.3.3 Name()

```
const char* FCFSupport::Owner::Name ( ) const [inline]
```

Return this owner's name.

References name.

### 9.104.3.4 operator=()

```
Owner& FCFSupport::Owner::operator= (
    Owner & other ) [inline]
```

The Assignment operator copies the contents of another [Owner](#) to this one.

#### Parameters

<i>other</i>	The other <a href="#">Owner</a> object.
--------------	---

References comment, initials, and name.

## 9.104.4 Member Data Documentation

### 9.104.4.1 comment

```
string FCFSupport::Owner::comment [private]
```

Commentary about this owner.

Referenced by [Comment\(\)](#), [operator=\(\)](#), and [Owner\(\)](#).

### 9.104.4.2 initials

```
string FCFSupport::Owner::initials [private]
```

This owner's initials.

Referenced by [Initials\(\)](#), [operator=\(\)](#), and [Owner\(\)](#).

#### 9.104.4.3 name

```
string FCFSupport::Owner::name [private]
```

This owner's name.

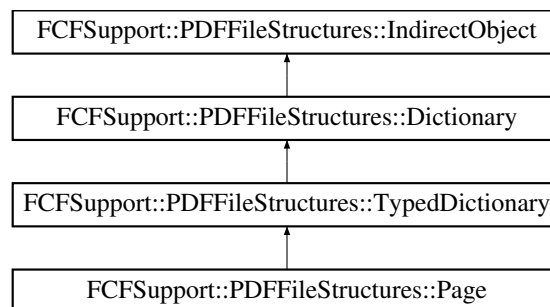
Referenced by Name(), operator=(), and Owner().

### 9.105 FCFSupport::PDFFileStructures::Page Class Reference

Describes a single page.

```
#include <PDFPrinterSupport.h>
```

Inheritance diagram for FCFSupport::PDFFileStructures::Page:



#### Public Member Functions

- [Page](#) ([ResourceDictionary](#) \*r=NULL, [Rectangle](#) \*mBox=NULL, [Rectangle](#) \*cBox=NULL, unsigned long int obj←Num=0L, unsigned short int genNum=0, [CrossReferenceTable](#) \*tab=NULL)  
*Constructor.*
- [~Page](#) ()  
*Destructor.*
- void [AppendStream](#) ([PDFStream](#) \*s)  
*Append a stream to the page.*

#### Protected Member Functions

- virtual ostream & [WriteDictionaryElements](#) (ostream &stream) const  
*Write an object directly.*



## Private Attributes

- [PageTree](#) \* [parent](#)  
*The page's parent page tree.*
- [ResourceDictionary](#) \* [resources](#)  
*The page's resources.*
- [Rectangle](#) \* [mediaBox](#)  
*This page's media box.*
- [Rectangle](#) \* [cropBox](#)  
*This page's crop box.*
- [PDFStreamVector](#) [contents](#)  
*This page's contents vector.*

## Friends

- class [PageTree](#)

### 9.105.1 Detailed Description

Describes a single page.

### 9.105.2 Constructor & Destructor Documentation

#### 9.105.2.1 Page()

```
FCFSupport::PDFFileStructures::Page::Page (
    ResourceDictionary * r = NULL,
    Rectangle * mBox = NULL,
    Rectangle * cBox = NULL,
    unsigned long int objNum = 0L,
    unsigned short int genNum = 0,
    CrossReferenceTable * tab = NULL ) [inline]
```

Constructor.

Create a fresh [Page](#) object.

#### Parameters

<i>r</i>	Resource <a href="#">Dictionary</a> .
<i>mBox</i>	Media box.
<i>cBox</i>	Crop box.
<i>objNum</i>	The next object number.
<i>genNum</i>	The generation number.
<i>tab</i>	The cross reference table we are in.

#### 9.105.2.2 ~Page()

```
FCFSupport::PDFFileStructures::Page::~~Page ( ) [inline]
```

Destructor.

### 9.105.3 Member Function Documentation

#### 9.105.3.1 AppendStream()

```
void FCFSupport::PDFFileStructures::Page::AppendStream (
    PDFStream * s ) [inline]
```

Append a stream to the page.

##### Parameters

<i>s</i>	The stream to append.
----------	-----------------------

References [lcc::stream](#).

#### 9.105.3.2 WriteDictionaryElements()

```
virtual ostream& FCFSupport::PDFFileStructures::Page::WriteDictionaryElements (
    ostream & stream ) const [protected], [virtual]
```

Write an object directly.

##### Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

Reimplemented from [FCFSupport::PDFFileStructures::TypedDictionary](#).

### 9.105.4 Friends And Related Function Documentation

#### 9.105.4.1 PageTree

```
friend class PageTree [friend]
```

### 9.105.5 Member Data Documentation

#### 9.105.5.1 contents

```
PDFStreamVector FCFSupport::PDFFileStructures::Page::contents [private]
```

This page's contents vector.

#### 9.105.5.2 cropBox

```
Rectangle* FCFSupport::PDFFileStructures::Page::cropBox [private]
```

This page's crop box.

#### 9.105.5.3 mediaBox

```
Rectangle* FCFSupport::PDFFileStructures::Page::mediaBox [private]
```

This page's media box.

#### 9.105.5.4 parent

```
PageTree* FCFSupport::PDFFileStructures::Page::parent [private]
```

The page's parent page tree.

#### 9.105.5.5 resources

```
ResourceDictionary* FCFSupport::PDFFileStructures::Page::resources [private]
```

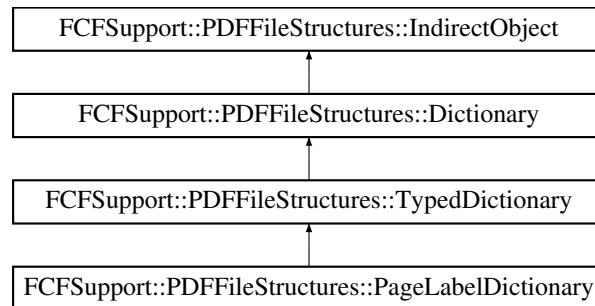
The page's resources.

## 9.106 FCFSupport::PDFFileStructures::PageLabelDictionary Class Reference

[Page](#) label dictionary.

```
#include <PDFPrinterSupport.h>
```

Inheritance diagram for FCFSupport::PDFFileStructures::PageLabelDictionary:



### Public Types

- enum [NumberStyle](#) {  
[None](#) = 0, [Decimal](#) = 'D', [UpperRoman](#) = 'R', [LowerRoman](#) = 'r',  
[UpperLetters](#) = 'A', [LowerLetters](#) = 'a' }

*Numbering style.*

### Public Member Functions

- [PageLabelDictionary](#) ([NumberStyle](#) s=[None](#), const string p="", int st=1, unsigned long int objNum=0L, unsigned short int genNum=0, CrossReferenceTable \*tab=NULL)

*Constructor.*

- [~PageLabelDictionary](#) ()

*Destructor.*

### Protected Member Functions

- virtual ostream & [WriteDictionaryElements](#) (ostream &stream) const

*Write an object directly.*

### Private Attributes

- [NumberStyle](#) style  
[Page](#) numbering style.
- string prefix  
Prefix string.
- int start  
[Page](#) numbering start.

### 9.106.1 Detailed Description

Page label dictionary.

Author

Robert Heller <heller@deepsoft.com>

### 9.106.2 Member Enumeration Documentation

#### 9.106.2.1 NumberStyle

```
enum FCFSupport::PDFFileStructures::PageLabelDictionary::NumberStyle
```

Numbering style.

Enumerator

None	None.
Decimal	Decimal arabic numerals.
UpperRoman	Uppercase roman numerals.
LowerRoman	Lowercase roman numerals.
UpperLetters	Uppercase letters.
LowerLetters	Lowercase letters.

### 9.106.3 Constructor & Destructor Documentation

#### 9.106.3.1 PageLabelDictionary()

```
FCFSupport::PDFFileStructures::PageLabelDictionary::PageLabelDictionary (
    NumberStyle s = None,
    const string p = "",
    int st = 1,
    unsigned long int objNum = 0L,
    unsigned short int genNum = 0,
    CrossReferenceTable * tab = NULL ) [inline]
```

Constructor.

Create a fresh Pager object.

## Parameters

<i>s</i>	Numbering style.
<i>p</i>	<a href="#">Page</a> label prefix string.
<i>st</i>	<a href="#">Page</a> number starting value for this range.
<i>objNum</i>	The next object number.
<i>genNum</i>	The generation number.
<i>tab</i>	The cross reference table we are in.

## 9.106.3.2 ~PageLabelDictionary()

```
FCFSupport::PDFFileStructures::PageLabelDictionary::~~PageLabelDictionary ( ) [inline]
```

Destructor.

References lcc::stream.

## 9.106.4 Member Function Documentation

## 9.106.4.1 WriteDictionaryElements()

```
virtual ostream& FCFSupport::PDFFileStructures::PageLabelDictionary::WriteDictionaryElements (
    ostream & stream ) const [protected], [virtual]
```

Write an object directly.

## Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

Reimplemented from [FCFSupport::PDFFileStructures::TypedDictionary](#).

## 9.106.5 Member Data Documentation

## 9.106.5.1 prefix

```
string FCFSupport::PDFFileStructures::PageLabelDictionary::prefix [private]
```

Prefix string.

## 9.106.5.2 start

```
int FCFSupport::PDFFileStructures::PageLabelDictionary::start [private]
```

Page numbering start.

## 9.106.5.3 style

```
NumberStyle FCFSupport::PDFFileStructures::PageLabelDictionary::style [private]
```

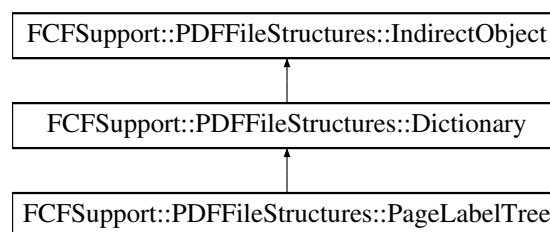
Page numbering style.

## 9.107 FCFSupport::PDFFileStructures::PageLabelTree Class Reference

A tree of page label dictionaries.

```
#include <PDFPrinterSupport.h>
```

Inheritance diagram for FCFSupport::PDFFileStructures::PageLabelTree:



## Public Member Functions

- [PageLabelTree](#) (unsigned long int objNum=0L, unsigned short int genNum=0, CrossReferenceTable \*tab=NULL)  
*Constructor.*
- [~PageLabelTree](#) ()  
*Destructor.*
- void [AddPageLabelTree](#) ([PageLabelTree](#) \*node)  
*Add a page label tree node.*
- void [AddPageLabelDictionary](#) (int number, [PageLabelDictionary](#) \*pld)  
*Add a page label dictionary.*
- int [Size](#) () const  
*Return the number of sub-nodes in this page label tree.*

## Protected Member Functions

- virtual ostream & [WriteDictionaryElements](#) (ostream &stream) const  
*Write an object directly.*

## Private Member Functions

- void [GetKidLimits](#) (int &lower, int &upper) const  
*Get limits of the kids vector.*

## Private Attributes

- bool [isRoot](#)  
*Root flag.*
- [PageLabelTreeKidVector](#) kids  
*Kid nodes.*
- [PageLabelDictionaryNumMap](#) nums  
*Num nodes.*

### 9.107.1 Detailed Description

A tree of page label dictionaries.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.107.2 Constructor & Destructor Documentation

#### 9.107.2.1 PageLabelTree()

```
FCFSSupport::PDFFileStructures::PageLabelTree::PageLabelTree (  
    unsigned long int objNum = 0L,  
    unsigned short int genNum = 0,  
    CrossReferenceTable * tab = NULL ) [inline]
```

Constructor.

Create a new page label tree.



## Parameters

<i>objNum</i>	The next object number.
<i>genNum</i>	The generation number.
<i>tab</i>	The cross reference table we are in.

## 9.107.2.2 ~PageLabelTree()

```
FCFSupport::PDFFileStructures::PageLabelTree::~~PageLabelTree ( ) [inline]
```

Destructor.

## 9.107.3 Member Function Documentation

## 9.107.3.1 AddPageLabelDictionary()

```
void FCFSupport::PDFFileStructures::PageLabelTree::AddPageLabelDictionary (
    int number,
    PageLabelDictionary * pld ) [inline]
```

Add a page label dictionary.

## Parameters

<i>number</i>	The page label dictionary start page number.
<i>pld</i>	<a href="#">Page</a> label dictionary pointer.

## 9.107.3.2 AddPageLabelTree()

```
void FCFSupport::PDFFileStructures::PageLabelTree::AddPageLabelTree (
    PageLabelTree * node ) [inline]
```

Add a page label tree node.

## Parameters

<i>node</i>	The page label tree node.
-------------	---------------------------

References `isRoot`.

Referenced by `FCFSupport::PDFFileStructures::CatalogDictionary::AddPageLabelTree()`.

#### 9.107.3.3 GetKidLimits()

```
void FCFSupport::PDFFileStructures::PageLabelTree::GetKidLimits (
    int & lower,
    int & upper ) const [private]
```

Get limits of the kids vector.

##### Parameters

<i>lower</i>	Lower end.
<i>upper</i>	Upper end.

#### 9.107.3.4 Size()

```
int FCFSupport::PDFFileStructures::PageLabelTree::Size ( ) const [inline]
```

Return the number of sub-nodes in this page label tree.

References `lcc::stream`.

#### 9.107.3.5 WriteDictionaryElements()

```
virtual ostream& FCFSupport::PDFFileStructures::PageLabelTree::WriteDictionaryElements (
    ostream & stream ) const [protected], [virtual]
```

Write an object directly.

##### Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

Reimplemented from [FCFSupport::PDFFileStructures::Dictionary](#).

### 9.107.4 Member Data Documentation

#### 9.107.4.1 isRoot

`bool FCFSupport::PDFFileStructures::PageLabelTree::isRoot [private]`

Root flag.

Referenced by `AddPageLabelTree()`.

#### 9.107.4.2 kids

`PageLabelTreeKidVector FCFSupport::PDFFileStructures::PageLabelTree::kids [private]`

Kid nodes.

#### 9.107.4.3 nums

`PageLabelDictionaryNumMap FCFSupport::PDFFileStructures::PageLabelTree::nums [private]`

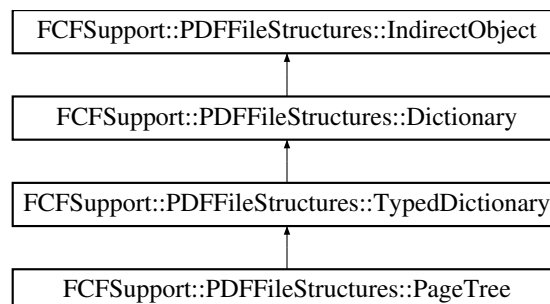
Num nodes.

## 9.108 FCFSupport::PDFFileStructures::PageTree Class Reference

A tree of pages.

```
#include <PDFPrinterSupport.h>
```

Inheritance diagram for `FCFSupport::PDFFileStructures::PageTree`:



## Public Member Functions

- [PageTree](#) ([ResourceDictionary](#) \*r=NULL, [Rectangle](#) \*mBox=NULL, [Rectangle](#) \*cBox=NULL, unsigned long int objNum=0L, unsigned short int genNum=0, [CrossReferenceTable](#) \*tab=NULL)

*Constructor.*

- [~PageTree](#) ()

*Destructor.*

- void [AddPage](#) ([Page](#) \*thepage)

*Add a page.*

- void [AddPageTree](#) ([PageTree](#) \*thepagetree)

*Add a tree of pages.*

## Protected Member Functions

- virtual ostream & [WriteDictionaryElements](#) (ostream &stream) const

*Write an object directly.*

## Private Attributes

- [PageTree](#) \* [parent](#)

*This page tree's parent.*

- [ResourceDictionary](#) \* [resources](#)

*Resources for this page tree.*

- [Rectangle](#) \* [mediaBox](#)

*Media box for this page tree.*

- [Rectangle](#) \* [cropBox](#)

*Crop box for this page tree.*

- vector< [TypedDictionary](#) \* > [pagenodes](#)

*The children of this page tree node.*

### 9.108.1 Detailed Description

A tree of pages.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.108.2 Constructor & Destructor Documentation

## 9.108.2.1 PageTree()

```
FCFSupport::PDFFileStructures::PageTree::PageTree (
    ResourceDictionary * r = NULL,
    Rectangle * mBox = NULL,
    Rectangle * cBox = NULL,
    unsigned long int objNum = 0L,
    unsigned short int genNum = 0,
    CrossReferenceTable * tab = NULL ) [inline]
```

Constructor.

Create a fresh Pager object.

## Parameters

<i>r</i>	Resource Dictionary.
<i>mBox</i>	Media box.
<i>cBox</i>	Crop box.
<i>objNum</i>	The next object number.
<i>genNum</i>	The generation number.
<i>tab</i>	The cross reference table we are in.

## 9.108.2.2 ~PageTree()

```
FCFSupport::PDFFileStructures::PageTree::~~PageTree ( ) [inline]
```

Destructor.

## 9.108.3 Member Function Documentation

## 9.108.3.1 AddPage()

```
void FCFSupport::PDFFileStructures::PageTree::AddPage (
    Page * thepage ) [inline]
```

Add a page.

## Parameters

<i>thepage</i>	The page to add.
----------------	------------------

### 9.108.3.2 AddPageTree()

```
void FCFSupport::PDFFileStructures::PageTree::AddPageTree (
    PageTree * thepagetree ) [inline]
```

Add a tree of pages.

#### Parameters

<i>thepagetree</i>	The page tree to add.
--------------------	-----------------------

References [lcc::stream](#).

Referenced by [FCFSupport::PDFFileStructures::CatalogDictionary::AddPageTree\(\)](#).

### 9.108.3.3 WriteDictionaryElements()

```
virtual ostream& FCFSupport::PDFFileStructures::PageTree::WriteDictionaryElements (
    ostream & stream ) const [protected], [virtual]
```

Write an object directly.

#### Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

Reimplemented from [FCFSupport::PDFFileStructures::TypedDictionary](#).

## 9.108.4 Member Data Documentation

### 9.108.4.1 cropBox

```
Rectangle* FCFSupport::PDFFileStructures::PageTree::cropBox [private]
```

Crop box for this page tree.

#### 9.108.4.2 mediaBox

```
Rectangle* FCFSupport::PDFFileStructures::PageTree::mediaBox [private]
```

Media box for this page tree.

#### 9.108.4.3 pagenodes

```
vector<TypedDictionary *> FCFSupport::PDFFileStructures::PageTree::pagenodes [private]
```

The children of this page tree node.

#### 9.108.4.4 parent

```
PageTree* FCFSupport::PDFFileStructures::PageTree::parent [private]
```

This page tree's parent.

#### 9.108.4.5 resources

```
ResourceDictionary* FCFSupport::PDFFileStructures::PageTree::resources [private]
```

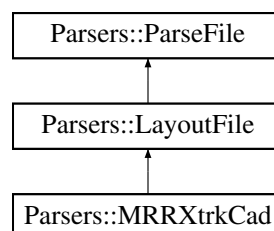
Resources for this page tree.

## 9.109 Parsers::ParseFile Class Reference

Virtual base class for file-based parsers.

```
#include <ParseFile.h>
```

Inheritance diagram for Parsers::ParseFile:



## Public Member Functions

- const char \* [SourceFile](#) () const  
*Return the name of the source file.*
- [ParseFile](#) (const char \*filename)  
*Constructor.*
- virtual [~ParseFile](#) ()  
*Destructor.*
- int [ProcessFile](#) (ostream &err)  
*open file and parse it.*

## Protected Member Functions

- virtual int [Parse](#) ()=0  
*The parser itself, supplied by derived classes.*
- virtual void [ParseError](#) (const char \*)=0  
*The parser's error handler, supplied by derived classes.*

## Protected Attributes

- char \* [lp](#)  
*Input line buffer pointer.*
- FILE \* [fp](#)  
*Input file pointer.*
- int [source\\_line](#)  
*Source line number.*
- char [line\\_buffer](#) [[buffersize](#)]  
*Input line buffer.*
- ostream \* [errorstream](#)  
*Stream for error reporting.*
- char \* [source\\_file](#)  
*Name of the source file.*

## Static Protected Attributes

- static const int [buffersize](#) = 1024  
*Size of line buffer.*

### 9.109.1 Detailed Description

Virtual base class for file-based parsers.

Contains all of the base level input and error output support members.

#### Author

Robert Heller <heller@deepsoft.com>



## 9.109.2 Constructor & Destructor Documentation

### 9.109.2.1 ParseFile()

```
Parsers::ParseFile::ParseFile (
    const char * filename ) [inline]
```

Constructor.

Make a local copy of the source file name, Other members are initialized.

### 9.109.2.2 ~ParseFile()

```
virtual Parsers::ParseFile::~~ParseFile ( ) [inline], [virtual]
```

Destructor.

Free up memory.

References ProcessFile(), and source\_file.

## 9.109.3 Member Function Documentation

### 9.109.3.1 Parse()

```
virtual int Parsers::ParseFile::Parse ( ) [protected], [pure virtual]
```

The parser itself, supplied by derived classes.

Implemented in [Parsers::LayoutFile](#).

### 9.109.3.2 ParseError()

```
virtual void Parsers::ParseFile::ParseError (
    const char * ) [protected], [pure virtual]
```

The parser's error handler, supplied by derived classes.

Implemented in [Parsers::LayoutFile](#).

### 9.109.3.3 ProcessFile()

```
int Parsers::ParseFile::ProcessFile (
    ostream & err )
```

open file and parse it.

**Parameters**

<i>err</i>	Output string to write error messages to.
------------	---

Referenced by `~ParseFile()`.

**9.109.3.4 SourceFile()**

```
const char* Parsers::ParseFile::SourceFile ( ) const [inline]
```

Return the name of the source file.

References `source_file`.

**9.109.4 Member Data Documentation****9.109.4.1 buffersize**

```
const int Parsers::ParseFile::buffersize = 1024 [static], [protected]
```

Size of line buffer.

**9.109.4.2 errorstream**

```
ostream* Parsers::ParseFile::errorstream [protected]
```

Stream for error reporting.

**9.109.4.3 fp**

```
FILE* Parsers::ParseFile::fp [protected]
```

Input file pointer.

#### 9.109.4.4 line\_buffer

```
char Parsers::ParseFile::line_buffer[buffer_size] [protected]
```

Input line buffer.

#### 9.109.4.5 lp

```
char* Parsers::ParseFile::lp [protected]
```

Input line buffer pointer.

#### 9.109.4.6 source\_file

```
char* Parsers::ParseFile::source_file [protected]
```

Name of the source file.

Referenced by SourceFile(), and ~ParseFile().

#### 9.109.4.7 source\_line

```
int Parsers::ParseFile::source_line [protected]
```

Source line number.

Used for error reporting.

## 9.110 ParseXML Class Reference

Class to hold an XML tree.

### Public Member Functions

- [ParseXML](#) (name, xml,...)  
*The constructor parses the XML string and stores it as a child of the rootnode component.*
- [displayTree](#) (fp="stdout")  
*Display the XML tree.*

## Static Public Member Functions

- static [validate](#) (object)  
*Validation typemethod.*

## Private Member Functions

- [\\_elementstart](#) (tag, attrlist,...)  
*Callback called at the start of of XML element.*
- [\\_elementend](#) (tag,...)  
*Callback called at the end of an XML element.*
- [\\_characterdata](#) (data)  
*Callback called with the text enclosed by an element.*

## Private Attributes

- [rootnode](#)  
*The (dummy) root node.*
- [nodeStack](#)  
*Temp variable used during parsing.*

### 9.110.1 Detailed Description

Class to hold an XML tree.

This class parses an XML string and stores the result as a DOM Element tree.

#### Parameters

<i>name</i>	Generally %%AUTO%% is passed.
<i>xml</i>	The XML string.
<i>—</i>	Options. None at present.

### 9.110.2 Constructor & Destructor Documentation

#### 9.110.2.1 ParseXML()

```
ParseXML::ParseXML (
    name ,
```

```
xml ,  
... )
```

The constructor parses the XML string and stores it as a child of the rootnode component.

**Parameters**

<i>xml</i>	The XML string.
<i>_</i>	Options. None at present.

**9.110.3 Member Function Documentation****9.110.3.1 `_characterdata()`**

```
ParseXML::_characterdata (  
    data ) [private]
```

Callback called with the text enclosed by an element.

**Parameters**

<i>data</i>	The text enclosed by an element. puts stderr "*** \$self _characterdata: nodeStack = \$nodeStack"
-------------	---

**9.110.3.2 `_elementend()`**

```
ParseXML::_elementend (  
    tag ,  
    ... ) [private]
```

Callback called at the end of an XML element.

**Parameters**

<i>tag</i>	The element's tag.
<i>←</i> <i>_←</i>	The element's options.

**9.110.3.3 `_elementstart()`**

```
ParseXML::_elementstart (  
    tag ,
```

```
    attrlist ,  
    ... ) [private]
```

Callback called at the start of of XML element.

#### Parameters

<i>tag</i>	The element's tag.
<i>attrlist</i>	The element's attribute list.
—	The element's options.

#### 9.110.3.4 displayTree()

```
ParseXML::displayTree (  
    fp = "stdout" )
```

Display the XML tree.

#### Parameters

<i>fp</i>	The channel to write the display to.
-----------	--------------------------------------

#### 9.110.3.5 validate()

```
static ParseXML::validate (  
    object ) [static]
```

Validation typemethod.

Raises an error if its argument is not a [ParseXML](#) object.

#### Parameters

<i>object</i>	The object to typecheck.
---------------	--------------------------

#### Returns

The object or raise an error.

### 9.110.4 Member Data Documentation

#### 9.110.4.1 nodeStack

```
ParseXML::nodeStack [private]
```

Temp variable used during parsing.

#### 9.110.4.2 rootnode

```
ParseXML::rootnode [private]
```

The (dummy) root node.

### 9.111 FCFSupport::PathName Class Reference

A Class that portably represents a pathname.

```
#include <PathName.h>
```

#### Public Member Functions

- [PathName](#) ()  
*Default constructor.*
- [PathName](#) (const char \*p)  
*Constructor, given a plain C string.*
- [PathName](#) (string p)  
*Constructor, given a STL basic\_string.*
- [PathName](#) (const [PathName](#) &other)  
*Copy constructor.*
- [~PathName](#) ()  
*Destructor.*
- [PathName](#) & [operator=](#) ([PathName](#) other)  
*Assignment operator, from another pathname.*
- [PathName](#) & [operator=](#) (string name)  
*Assignment operator, from a string.*
- bool [operator==](#) (const [PathName](#) other) const  
*Equality operator.*
- bool [operator<](#) (const [PathName](#) other) const  
*Less than operator.*
- bool [operator>](#) (const [PathName](#) other) const  
*Greater than operator.*
- bool [operator<=](#) (const [PathName](#) other) const  
*Less than or equal operator.*
- bool [operator>=](#) (const [PathName](#) other) const



*Greater than or equal operator.*

- bool [SameDirectory](#) (const [PathName](#) other) const

*Are the two pathnames in the same directory?*

- string [Tail](#) () const

*Return the last pathname component.*

- string [Dirname](#) () const

*Return only the directory name.*

- string [Extension](#) () const

*Return only the extension.*

- string [FullPath](#) () const

*Return the full pathname.*

- [stringVector Split](#) () const

*Return a list of pathname components.*

- char [PathSeparator](#) () const

*Return the pathname separator character.*

- [PathName operator+](#) (const [PathName](#) other)

*Concatenate pathnames.*

- [PathName operator+](#) (string tail)

*Concatenate a string to the tail of a pathname.*

- [PathName & operator+=](#) (const [PathName](#) other)

*Append a pathname.*

- [PathName & operator+=](#) (string tail)

*Append a string.*

## Private Attributes

- string [pathname](#)

*The pathname string.*

### 9.111.1 Detailed Description

A Class that portably represents a pathname.

This class implements a pathname object in a portable, cross platform way.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.111.2 Constructor & Destructor Documentation

**9.111.2.1 PathName()** [1/4]

```
FCFSupport::PathName::PathName ( ) [inline]
```

Default constructor.

References pathname.

**9.111.2.2 PathName()** [2/4]

```
FCFSupport::PathName::PathName (
    const char * p ) [inline]
```

Constructor, given a plain C string.

**Parameters**

<i>p</i>	The plain C string.
----------	---------------------

References pathname.

**9.111.2.3 PathName()** [3/4]

```
FCFSupport::PathName::PathName (
    string p ) [inline]
```

Constructor, given a STL basic\_string.

**Parameters**

<i>p</i>	The STL basic_string.
----------	-----------------------

References pathname.

**9.111.2.4 PathName()** [4/4]

```
FCFSupport::PathName::PathName (
    const PathName & other ) [inline]
```

Copy constructor.

## Parameters

<i>other</i>	The other instance.
--------------	---------------------

References pathname.

## 9.111.2.5 ~PathName()

```
FCFSupport::PathName::~~PathName ( ) [inline]
```

Destructor.

## 9.111.3 Member Function Documentation

## 9.111.3.1 Dirname()

```
string FCFSupport::PathName::Dirname ( ) const
```

Return only the directory name.

Referenced by SameDirectory().

## 9.111.3.2 Extension()

```
string FCFSupport::PathName::Extension ( ) const
```

Return only the extension.

Referenced by SameDirectory().

## 9.111.3.3 FullPath()

```
string FCFSupport::PathName::FullPath ( ) const [inline]
```

Return the full pathname.

References operator+(), operator+=(), pathname, PathSeparator(), and Split().

Referenced by FCFSupport::System::CarsFile(), FCFSupport::System::CarTypesFile(), FCFSupport::System::IndustriesFile(), FCFSupport::System::OrdersFile(), FCFSupport::System::OwnersFile(), FCFSupport::System::StatsFile(), FCFSupport::System::SystemFile(), and FCFSupport::System::TrainsFile().

## 9.111.3.4 operator+() [1/2]

```
PathName FCFSupport::PathName::operator+ (
    const PathName other )
```

Concatenate pathnames.

**Parameters**

<i>other</i>	The other instance.
--------------	---------------------

Referenced by FullPath().

**9.111.3.5 operator+()** [2/2]

```
PathName FCFSupport::PathName::operator+ (  
    string tail )
```

Concatenate a string to the tail of a pathname.

**Parameters**

<i>tail</i>	The STL basic_string.
-------------	-----------------------

**9.111.3.6 operator+=()** [1/2]

```
PathName& FCFSupport::PathName::operator+= (  
    const PathName other )
```

Append a pathname.

**Parameters**

<i>other</i>	The other instance.
--------------	---------------------

Referenced by FullPath().

**9.111.3.7 operator+=()** [2/2]

```
PathName& FCFSupport::PathName::operator+= (  
    string tail )
```

Append a string.

## Parameters

<i>tail</i>	The STL basic_string.
-------------	-----------------------

## 9.111.3.8 operator&lt;()

```
bool FCFSupport::PathName::operator< (
    const PathName other ) const [inline]
```

Less than operator.

## Parameters

<i>other</i>	The other instance.
--------------	---------------------

References pathname.

## 9.111.3.9 operator&lt;=()

```
bool FCFSupport::PathName::operator<= (
    const PathName other ) const [inline]
```

Less than or equal operator.

## Parameters

<i>other</i>	The other instance.
--------------	---------------------

References pathname.

## 9.111.3.10 operator=() [1/2]

```
PathName& FCFSupport::PathName::operator= (
    PathName other ) [inline]
```

Assignment operator, from another pathname.

**Parameters**

<i>other</i>	The other instance.
--------------	---------------------

References pathname.

**9.111.3.11 operator=()** [2/2]

```
PathName& FCFSupport::PathName::operator= (
    string name ) [inline]
```

Assignment operator, from a string.

**Parameters**

<i>name</i>	The STL basic_string.
-------------	-----------------------

References pathname.

**9.111.3.12 operator==()**

```
bool FCFSupport::PathName::operator== (
    const PathName other ) const [inline]
```

Equality operator.

**Parameters**

<i>other</i>	The other instance.
--------------	---------------------

References pathname.

**9.111.3.13 operator>()**

```
bool FCFSupport::PathName::operator> (
    const PathName other ) const [inline]
```

Greater than operator.

**Parameters**

<i>other</i>	The other instance.
--------------	---------------------

References pathname.

**9.111.3.14 operator>=()**

```
bool FCFSupport::PathName::operator>= (
    const PathName other ) const [inline]
```

Greater than or equal operator.

**Parameters**

<i>other</i>	The other instance.
--------------	---------------------

References pathname.

**9.111.3.15 PathSeparator()**

```
char FCFSupport::PathName::PathSeparator ( ) const
```

Return the pathname separator character.

Referenced by FullPath().

**9.111.3.16 SameDirectory()**

```
bool FCFSupport::PathName::SameDirectory (
    const PathName other ) const [inline]
```

Are the two pathnames in the same directory?

**Parameters**

<i>other</i>	The other instance.
--------------	---------------------

References Dirname(), Extension(), and Tail().

#### 9.111.3.17 Split()

```
stringVector FCFSupport::PathName::Split ( ) const
```

Return a list of pathname components.

Referenced by FullPath().

#### 9.111.3.18 Tail()

```
string FCFSupport::PathName::Tail ( ) const
```

Return the last pathname component.

Referenced by SameDirectory().

### 9.111.4 Member Data Documentation

#### 9.111.4.1 pathname

```
string FCFSupport::PathName::pathname [private]
```

The pathname string.

Referenced by FullPath(), operator<(), operator<=(), operator=(), operator==(), operator>(), operator>=(), and Path↔Name().

## 9.112 TTSupport::PathName Class Reference

A Class that portably represents a pathname.

```
#include <PathName.h>
```



## Public Member Functions

- [PathName](#) ()  
*Default constructor.*
- [PathName](#) (const char \*p)  
*Constructor, given a plain C string.*
- [PathName](#) (string p)  
*Constructor, given a STL basic\_string.*
- [PathName](#) (const [PathName](#) &other)  
*Copy constructor.*
- [~PathName](#) ()  
*Destructor.*
- [PathName](#) & [operator=](#) ([PathName](#) other)  
*Assignment operator, from another pathname.*
- [PathName](#) & [operator=](#) (string name)  
*Assignment operator, from a string.*
- bool [operator==](#) (const [PathName](#) other) const  
*Equality operator.*
- bool [operator<](#) (const [PathName](#) other) const  
*Less than operator.*
- bool [operator>](#) (const [PathName](#) other) const  
*Greater than operator.*
- bool [operator<=](#) (const [PathName](#) other) const  
*Less than or equal operator.*
- bool [operator>=](#) (const [PathName](#) other) const  
*Greater than or equal operator.*
- bool [SameDirectory](#) (const [PathName](#) other) const  
*Are the two pathnames in the same directory?*
- string [Tail](#) () const  
*Return the last pathname component.*
- string [Dirname](#) () const  
*Return only the directory name.*
- string [Extension](#) () const  
*Return only the extension.*
- string [FullPath](#) () const  
*Return the full pathname.*
- [stringVector](#) [Split](#) () const  
*Return a list of pathname components.*
- char [PathSeparator](#) () const  
*Return the pathname separator character.*
- [PathName](#) [operator+](#) (const [PathName](#) other)  
*Concatenate pathnames.*
- [PathName](#) [operator+](#) (string tail)  
*Concatenate a string to the tail of a pathname.*
- [PathName](#) & [operator+=](#) (const [PathName](#) other)  
*Append a pathname.*
- [PathName](#) & [operator+=](#) (string tail)  
*Append a string.*

## Private Attributes

- string [pathname](#)

*The pathname string.*

### 9.112.1 Detailed Description

A Class that portably represents a pathname.

This class implements a pathname object in a portable, cross platform way.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.112.2 Constructor & Destructor Documentation

#### 9.112.2.1 PathName() [1/4]

```
TTSupport::PathName::PathName ( ) [inline]
```

Default constructor.

References [pathname](#).

#### 9.112.2.2 PathName() [2/4]

```
TTSupport::PathName::PathName (
    const char * p ) [inline]
```

Constructor, given a plain C string.

#### Parameters

<i>p</i>	The plain C string.
----------	---------------------

References [pathname](#).

### 9.112.2.3 PathName() [3/4]

```
TTSupport::PathName::PathName (
    string p ) [inline]
```

Constructor, given a STL basic\_string.

#### Parameters

<i>p</i>	The STL basic_string.
----------	-----------------------

References pathname.

### 9.112.2.4 PathName() [4/4]

```
TTSupport::PathName::PathName (
    const PathName & other ) [inline]
```

Copy constructor.

#### Parameters

<i>other</i>	The other instance.
--------------	---------------------

References pathname.

### 9.112.2.5 ~PathName()

```
TTSupport::PathName::~~PathName ( ) [inline]
```

Destructor.

## 9.112.3 Member Function Documentation

### 9.112.3.1 Dirname()

```
string TTSupport::PathName::Dirname ( ) const
```

Return only the directory name.

Referenced by SameDirectory().

#### 9.112.3.2 Extension()

```
string TTSupport::PathName::Extension ( ) const
```

Return only the extension.

Referenced by SameDirectory().

#### 9.112.3.3 FullPath()

```
string TTSupport::PathName::FullPath ( ) const [inline]
```

Return the full pathname.

References operator+(), operator+=( ), pathname, PathSeparator(), and Split().

#### 9.112.3.4 operator+() [1/2]

```
PathName TTSupport::PathName::operator+ (
    const PathName other )
```

Concatenate pathnames.

##### Parameters

<i>other</i>	The other instance.
--------------	---------------------

Referenced by FullPath().

#### 9.112.3.5 operator+() [2/2]

```
PathName TTSupport::PathName::operator+ (
    string tail )
```

Concatenate a string to the tail of a pathname.

##### Parameters

<i>tail</i>	The STL basic_string.
-------------	-----------------------

### 9.112.3.6 operator+=() [1/2]

```
PathName& TTSupport::PathName::operator+= (
    const PathName other )
```

Append a pathname.

#### Parameters

<i>other</i>	The other instance.
--------------	---------------------

Referenced by FullPath().

### 9.112.3.7 operator+=() [2/2]

```
PathName& TTSupport::PathName::operator+= (
    string tail )
```

Append a string.

#### Parameters

<i>tail</i>	The STL basic_string.
-------------	-----------------------

### 9.112.3.8 operator<()

```
bool TTSupport::PathName::operator< (
    const PathName other ) const [inline]
```

Less than operator.

#### Parameters

<i>other</i>	The other instance.
--------------	---------------------

References pathname.

### 9.112.3.9 operator<=()

```
bool TTSupport::PathName::operator<= (
    const PathName other ) const [inline]
```

Less than or equal operator.

#### Parameters

<i>other</i>	The other instance.
--------------	---------------------

References pathname.

### 9.112.3.10 operator=() [1/2]

```
PathName& TTSupport::PathName::operator= (
    PathName other ) [inline]
```

Assignment operator, from another pathname.

#### Parameters

<i>other</i>	The other instance.
--------------	---------------------

References pathname.

### 9.112.3.11 operator=() [2/2]

```
PathName& TTSupport::PathName::operator= (
    string name ) [inline]
```

Assignment operator, from a string.

#### Parameters

<i>name</i>	The STL basic_string.
-------------	-----------------------

References pathname.

### 9.112.3.12 operator==( )

```
bool TTSupport::PathName::operator== (
    const PathName other ) const [inline]
```

Equality operator.

#### Parameters

<i>other</i>	The other instance.
--------------	---------------------

References pathname.

### 9.112.3.13 operator>( )

```
bool TTSupport::PathName::operator> (
    const PathName other ) const [inline]
```

Greater than operator.

#### Parameters

<i>other</i>	The other instance.
--------------	---------------------

References pathname.

### 9.112.3.14 operator>=( )

```
bool TTSupport::PathName::operator>= (
    const PathName other ) const [inline]
```

Greater than or equal operator.

#### Parameters

<i>other</i>	The other instance.
--------------	---------------------

References pathname.

#### 9.112.3.15 PathSeparator()

```
char TTSupport::PathName::PathSeparator ( ) const
```

Return the pathname separator character.

Referenced by FullPath().

#### 9.112.3.16 SameDirectory()

```
bool TTSupport::PathName::SameDirectory (
    const PathName other ) const [inline]
```

Are the two pathnames in the same directory?

##### Parameters

<i>other</i>	The other instance.
--------------	---------------------

References Dirname(), Extension(), and Tail().

#### 9.112.3.17 Split()

```
stringVector TTSupport::PathName::Split ( ) const
```

Return a list of pathname components.

Referenced by FullPath().

#### 9.112.3.18 Tail()

```
string TTSupport::PathName::Tail ( ) const
```

Return the last pathname component.

Referenced by SameDirectory().

### 9.112.4 Member Data Documentation



## 9.112.4.1 pathname

```
string TTSupport::PathName::pathname [private]
```

The pathname string.

Referenced by FullPath(), operator<(), operator<=(), operator=(), operator==(), operator>(), operator>=(), and Path↵Name().

## 9.113 FCFSupport::PauseCallback Class Reference

The Pause callback.

```
#include <CallBack.h>
```

## Public Member Functions

- [PauseCallback](#) ()  
*The constructor.*
- virtual [~PauseCallback](#) ()  
*The destructor.*
- virtual void [Pause](#) (string message) const  
*Display a message and wait for a user response.*

## 9.113.1 Detailed Description

The Pause callback.

This callback displays a message and waits for a user response. There is no partituar response sought, just an acknowledgement to continue processing. Usually there is something the user should take a momment to check or read before proceeding.

```
@author Robert Heller \<heller\@deepsoft.com\>
```

## 9.113.2 Constructor &amp; Destructor Documentation

## 9.113.2.1 PauseCallback()

```
FCFSupport::PauseCallback::PauseCallback ( ) [inline]
```

The constructor.

The base constructor does nothing. It is presumed that a derived class might do something useful.

### 9.113.2.2 ~PauseCallback()

```
virtual FCFSupport::PauseCallback::~~PauseCallback ( ) [inline], [virtual]
```

The destructor.

The base destructor does nothing. It is presumed that a derived class might do something useful.

## 9.113.3 Member Function Documentation

### 9.113.3.1 Pause()

```
virtual void FCFSupport::PauseCallback::Pause (
    string message ) const [inline], [virtual]
```

Display a message and wait for a user response.

This message just displays a message and waits for a user response (acknowledgement).

#### Parameters

<i>message</i>	The message to display when pausing.
----------------	--------------------------------------

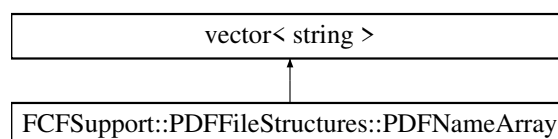
References `gettext::_()`, `FCFSupport::WorkInProgressCallback::ProgressDone()`, `FCFSupport::WorkInProgressCallback::ProgressStart()`, and `FCFSupport::WorkInProgressCallback::ProgressUpdate()`.

## 9.114 FCFSupport::PDFFileStructures::PDFNameArray Class Reference

PDF Name array.

```
#include <PDFPrinterSupport.h>
```

Inheritance diagram for `FCFSupport::PDFFileStructures::PDFNameArray`:



## Public Member Functions

- [PDFNameArray](#) ()  
*Constructor.*
- [~PDFNameArray](#) ()  
*Destructor.*

### 9.114.1 Detailed Description

PDF Name array.

Used with resource dictionaries.

Author

Robert Heller <heller@deepsoft.com>

### 9.114.2 Constructor & Destructor Documentation

#### 9.114.2.1 PDFNameArray()

```
FCFSupport::PDFFileStructures::PDFNameArray::PDFNameArray ( ) [inline]
```

Constructor.

#### 9.114.2.2 ~PDFNameArray()

```
FCFSupport::PDFFileStructures::PDFNameArray::~~PDFNameArray ( ) [inline]
```

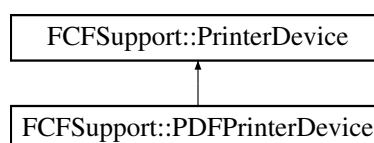
Destructor.

## 9.115 FCFSupport::PDFPrinterDevice Class Reference

PDF Printer device.

```
#include <PDFPrinter.h>
```

Inheritance diagram for FCFSupport::PDFPrinterDevice:



## Public Member Functions

- [PDFPrinterDevice](#) (const string filename="", const string title\_="", [PageSize](#) pageSize=[Letter](#), char \*\*outmessage=NULL)  
*Constructor.*
- virtual bool [OpenPrinter](#) (const string filename, [PageSize](#) pageSize=[Letter](#), char \*\*outmessage=NULL)  
*Open the printer file.*
- virtual bool [ClosePrinter](#) (char \*\*outmessage=NULL)  
*Close the printer.*
- virtual bool [SetTypeSpacing](#) ([TypeSpacing](#) spacing)  
*Set the type spacing.*
- virtual bool [SetTypeWeight](#) ([TypeWeight](#) weight)  
*Set the type weight.*
- virtual bool [SetTypeSlant](#) ([TypeSlant](#) slant)  
*Set the type slant.*
- virtual bool [NewPage](#) (const string heading="")  
*Generate a new page.*
- virtual bool [PutLine](#) (const string line="")  
*Put a line of text.*
- virtual bool [Put](#) (const string text)  
*Put a string.*
- virtual bool [Tab](#) (int column)  
*Move to the specified tab column.*
- virtual [~PDFPrinterDevice](#) ()  
*Destructor.*

## Private Member Functions

- bool [CreateNewPage](#) ()  
*Create a new page.*
- bool [CreateNewStream](#) ()  
*Create new stream.*

## Private Attributes

- [PDFFileStructures::CrossReferenceTable](#) crossReferenceTable  
*PDF Cross reference table.*
- [PDFFileStructures::CatalogDictionary](#) \* rootDictionary  
*PDF Root catalog.*
- [PDFFileStructures::Page](#) \* currentPage  
*Current PDF Page.*
- [PDFFileStructures::PDFStream](#) \* currentStream  
*Current PDF Stream.*
- [PDFFileStructures::PageTree](#) \* pageTreeRoot  
*PDF Page Tree root.*
- [PDFFileStructures::InformationDirectory](#) \* info

*Information dictionary.*

- ofstream [printerStream](#)

*Output stream.*

- string [title](#)

*Title string.*

- string [currentFontName](#)

*Current font name.*

- int [lines](#)

*Number of lines.*

- int [horizontalScaling](#)

*Current horizontal scaling.*

- int [maxLines](#)

*Maximum number of lines.*

- bool [partline](#)

*Partial line flag.*

- bool [needPage](#)

*Need page flag.*

- int [currentColumn](#)

*Current column.*

- double [currentColumnFraction](#)

*Current column fraction.*

## Additional Inherited Members

### 9.115.1 Detailed Description

PDF Printer device.

Author

Robert Heller <heller@deepsoft.com>

### 9.115.2 Constructor & Destructor Documentation

#### 9.115.2.1 PDFPrinterDevice()

```
FCFSupport::PDFPrinterDevice::PDFPrinterDevice (
    const string filename = "",
    const string title_ = "",
    PageSize pageSize = Letter,
    char ** outmessage = NULL )
```

Constructor.

Create a PDF Printer device.

## Parameters

<i>filename</i>	The name of the file to print to.
<i>title</i>	The document title.
<i>pageSize</i>	The document page size.
<i>outmessage</i>	Pointer to get an error message buffer pointer.

## 9.115.2.2 ~PDFPrinterDevice()

```
virtual FCFSupport::PDFPrinterDevice::~~PDFPrinterDevice ( ) [virtual]
```

Destructor.

## 9.115.3 Member Function Documentation

## 9.115.3.1 ClosePrinter()

```
virtual bool FCFSupport::PDFPrinterDevice::ClosePrinter (
    char ** outmessage = NULL ) [virtual]
```

Close the printer.

## Parameters

<i>outmessage</i>	Pointer to get an error message buffer pointer.
-------------------	---

Reimplemented from [FCFSupport::PrinterDevice](#).

## 9.115.3.2 CreateNewPage()

```
bool FCFSupport::PDFPrinterDevice::CreateNewPage ( ) [private]
```

Create a new page.

### 9.115.3.3 CreateNewStream()

```
bool FCFSupport::PDFPrinterDevice::CreateNewStream ( ) [private]
```

Create new stream.

### 9.115.3.4 NewPage()

```
virtual bool FCFSupport::PDFPrinterDevice::NewPage (
    const string heading = "" ) [virtual]
```

Generate a new page.

#### Parameters

<i>heading</i>	The new page heading string.
----------------	------------------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

### 9.115.3.5 OpenPrinter()

```
virtual bool FCFSupport::PDFPrinterDevice::OpenPrinter (
    const string filename,
    PageSize pageSize = Letter,
    char ** outmessage = NULL ) [virtual]
```

Open the printer file.

#### Parameters

<i>filename</i>	The name of the file to print to.
<i>pageSize</i>	The document page size.
<i>outmessage</i>	Pointer to get an error message buffer pointer.

Reimplemented from [FCFSupport::PrinterDevice](#).

### 9.115.3.6 Put()

```
virtual bool FCFSupport::PDFPrinterDevice::Put (
    const string text ) [virtual]
```

Put a string.

**Parameters**

<i>text</i>	The text string to print.
-------------	---------------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

#### 9.115.3.7 PutLine()

```
virtual bool FCFSupport::PDFPrinterDevice::PutLine (
    const string line = "" ) [virtual]
```

Put a line of text.

**Parameters**

<i>line</i>	The line of text.
-------------	-------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

#### 9.115.3.8 SetTypeSlant()

```
virtual bool FCFSupport::PDFPrinterDevice::SetTypeSlant (
    TypeSlant slant ) [virtual]
```

Set the type slant.

**Parameters**

<i>slant</i>	The slant value to set.
--------------	-------------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

#### 9.115.3.9 SetTypeSpacing()

```
virtual bool FCFSupport::PDFPrinterDevice::SetTypeSpacing (
    TypeSpacing spacing ) [virtual]
```

Set the type spacing.



## Parameters

<i>spacing</i>	The spacing value to set.
----------------	---------------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

### 9.115.3.10 SetTypeWeight()

```
virtual bool FCFSupport::PDFPrinterDevice::SetTypeWeight (
    TypeWeight weight ) [virtual]
```

Set the type weight.

## Parameters

<i>weight</i>	The weight value to set.
---------------	--------------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

### 9.115.3.11 Tab()

```
virtual bool FCFSupport::PDFPrinterDevice::Tab (
    int column ) [virtual]
```

Move to the specified tab column.

## Parameters

<i>column</i>	the column to move to.
---------------	------------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

## 9.115.4 Member Data Documentation

### 9.115.4.1 crossReferenceTable

```
PDFFileStructures::CrossReferenceTable FCFSupport::PDFPrinterDevice::crossReferenceTable [private]
```

PDF Cross reference table.

#### 9.115.4.2 currentColumn

```
int FCFSupport::PDFPrinterDevice::currentColumn [private]
```

Current column.

#### 9.115.4.3 currentColumnFraction

```
double FCFSupport::PDFPrinterDevice::currentColumnFraction [private]
```

Current column fraction.

#### 9.115.4.4 currentFontName

```
string FCFSupport::PDFPrinterDevice::currentFontName [private]
```

Current font name.

#### 9.115.4.5 currentPage

```
PDFFileStructures::Page* FCFSupport::PDFPrinterDevice::currentPage [private]
```

Current PDF Page.

#### 9.115.4.6 currentStream

```
PDFFileStructures::PDFStream* FCFSupport::PDFPrinterDevice::currentStream [private]
```

Current PDF Stream.

#### 9.115.4.7 horizontalScaling

```
int FCFSupport::PDFPrinterDevice::horizontalScaling [private]
```

Current horizontal scaling.

#### 9.115.4.8 info

`PDFFileStructures::InformationDirectory*` FCFSupport::PDFPrinterDevice::info [private]

Information dictionary.

#### 9.115.4.9 lines

`int` FCFSupport::PDFPrinterDevice::lines [private]

Number of lines.

#### 9.115.4.10 maxLines

`int` FCFSupport::PDFPrinterDevice::maxLines [private]

Maximum number of lines.

#### 9.115.4.11 needPage

`bool` FCFSupport::PDFPrinterDevice::needPage [private]

Need page flag.

#### 9.115.4.12 pageTreeRoot

`PDFFileStructures::PageTree*` FCFSupport::PDFPrinterDevice::pageTreeRoot [private]

PDF Page Tree root.

#### 9.115.4.13 partline

`bool` FCFSupport::PDFPrinterDevice::partline [private]

Partial line flag.

#### 9.115.4.14 printerStream

```
ofstream FCFSupport::PDFPrinterDevice::printerStream [private]
```

Output stream.

#### 9.115.4.15 rootDictionary

```
PDFFileStructures::CatalogDictionary* FCFSupport::PDFPrinterDevice::rootDictionary [private]
```

PDF Root catalog.

#### 9.115.4.16 title

```
string FCFSupport::PDFPrinterDevice::title [private]
```

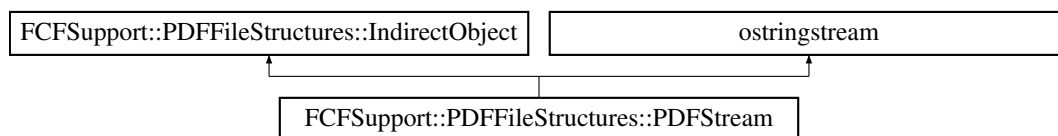
Title string.

### 9.116 FCFSupport::PDFFileStructures::PDFStream Class Reference

Stream object.

```
#include <PDFPrinterSupport.h>
```

Inheritance diagram for FCFSupport::PDFFileStructures::PDFStream:



#### Public Member Functions

- [PDFStream](#) (unsigned long int objNum=0L, unsigned short int genNum=0, CrossReferenceTable \*tab=NULL)  
*Constructor.*
- [~PDFStream](#) ()  
*Destructor.*
- virtual ostream & [WriteDirect](#) (ostream &stream) const  
*Write an object directly.*

### 9.116.1 Detailed Description

Stream object.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.116.2 Constructor & Destructor Documentation

#### 9.116.2.1 PDFStream()

```
FCFSupport::PDFFileStructures::PDFStream::PDFStream (
    unsigned long int objNum = 0L,
    unsigned short int genNum = 0,
    CrossReferenceTable * tab = NULL ) [inline]
```

Constructor.

Create a stream object.

#### Parameters

<i>objNum</i>	The next object number.
<i>genNum</i>	The generation number.
<i>tab</i>	The cross reference table we are in.

#### 9.116.2.2 ~PDFStream()

```
FCFSupport::PDFFileStructures::PDFStream::~~PDFStream ( ) [inline]
```

Destructor.

References `lcc::stream`.

### 9.116.3 Member Function Documentation

### 9.116.3.1 WriteDirect()

```
virtual ostream& FCFSupport::PDFFileStructures::PDFStream::WriteDirect (
    ostream & stream ) const [virtual]
```

Write an object directly.

#### Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

Implements [FCFSupport::PDFFileStructures::IndirectObject](#).

## 9.117 Parsers::TrackGraph::Point Struct Reference

Position structure.

### Public Attributes

- double [x](#)  
*X coordinate.*
- double [y](#)  
*Y coordinate.*

### 9.117.1 Detailed Description

Position structure.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.117.2 Member Data Documentation

#### 9.117.2.1 x

```
double Parsers::TrackGraph::Point::x
```

X coordinate.

Referenced by [Parsers::TrackGraph::CompressedNodeValues::CompressedNodeValues\(\)](#).

## 9.117.2.2 y

```
double Parsers::TrackGraph::Point::y
```

Y coordinate.

Referenced by `Parsers::TrackGraph::CompressedNodeValues::CompressedNodeValues()`.

## 9.118 Parsers::TurnoutBodyElt::Pos Struct Reference

Position structure.

### Public Attributes

- float [x](#)  
*\$X\$ coordinate.*
- float [y](#)  
*\$Y\$ coordinate.*

### 9.118.1 Detailed Description

Position structure.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.118.2 Member Data Documentation

## 9.118.2.1 x

```
float Parsers::TurnoutBodyElt::Pos::x
```

*\$X\$ coordinate.*

Referenced by `Parsers::TurnoutBodyElt::GetTurnoutCurveSegment()`, `Parsers::TurnoutBodyElt::GetTurnoutJointSegment()`, `Parsers::TurnoutBodyElt::GetTurnoutStraightSegment()`, `Parsers::TurnoutBodyElt::MakeTurnoutCurveSegment()`, `Parsers::TurnoutBodyElt::MakeTurnoutJointSegment()`, and `Parsers::TurnoutBodyElt::MakeTurnoutStraightSegment()`.

### 9.118.2.2 y

```
float Parsers::TurnoutBodyElt::Pos::y
```

\$Y\$ coordinate.

Referenced by `Parsers::TurnoutBodyElt::GetTurnoutCurveSegment()`, `Parsers::TurnoutBodyElt::GetTurnoutJointSegment()`, `Parsers::TurnoutBodyElt::GetTurnoutStraightSegment()`, `Parsers::TurnoutBodyElt::MakeTurnoutCurveSegment()`, `Parsers::TurnoutBodyElt::MakeTurnoutJointSegment()`, and `Parsers::TurnoutBodyElt::MakeTurnoutStraightSegment()`.

## 9.119 Parsers::BezierBodyElt::Pos Struct Reference

Position structure.

### Public Attributes

- float [x](#)  
*\$X\$ coordinate.*
- float [y](#)  
*\$Y\$ coordinate.*

### 9.119.1 Detailed Description

Position structure.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.119.2 Member Data Documentation

#### 9.119.2.1 x

```
float Parsers::BezierBodyElt::Pos::x
```

\$X\$ coordinate.

Referenced by `Parsers::BezierBodyElt::GetCurveSegment()`, `Parsers::BezierBodyElt::GetStraightSegment()`, `Parsers::BezierBodyElt::MakeCurveSegment()`, and `Parsers::BezierBodyElt::MakeStraightSegment()`.



## 9.119.2.2 y

```
float Parsers::BezierBodyElt::Pos::y
```

\$Y\$ coordinate.

Referenced by Parsers::BezierBodyElt::GetCurveSegment(), Parsers::BezierBodyElt::GetStraightSegment(), Parsers::BezierBodyElt::MakeCurveSegment(), and Parsers::BezierBodyElt::MakeStraightSegment().

## 9.120 Parsers::CornuBodyElt::Pos Struct Reference

Position structure.

### Public Attributes

- float [x](#)  
\$X\$ coordinate.
- float [y](#)  
\$Y\$ coordinate.

### 9.120.1 Detailed Description

Position structure.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.120.2 Member Data Documentation

## 9.120.2.1 x

```
float Parsers::CornuBodyElt::Pos::x
```

\$X\$ coordinate.

Referenced by Parsers::CornuBodyElt::GetCurveSegment(), Parsers::CornuBodyElt::GetStraightSegment(), Parsers::CornuBodyElt::MakeCurveSegment(), and Parsers::CornuBodyElt::MakeStraightSegment().

## 9.120.2.2 y

```
float Parsers::CornuBodyElt::Pos::y
```

\$Y\$ coordinate.

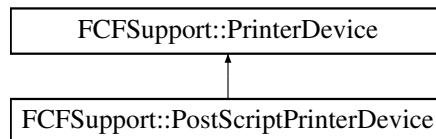
Referenced by `Parsers::CornuBodyElt::GetCurveSegment()`, `Parsers::CornuBodyElt::GetStraightSegment()`, `Parsers::CornuBodyElt::MakeCurveSegment()`, and `Parsers::CornuBodyElt::MakeStraightSegment()`.

## 9.121 FCFSupport::PostScriptPrinterDevice Class Reference

Derived class for printing on Postscript printers.

```
#include <PostScriptPrinter.h>
```

Inheritance diagram for `FCFSupport::PostScriptPrinterDevice`:



### Public Member Functions

- [PostScriptPrinterDevice](#) (const string filename="", const string title\_="", [PageSize](#) pageSize=[Letter](#), char \*\*outmessage=NULL)  
*Constructor.*
- virtual bool [OpenPrinter](#) (const string filename, [PageSize](#) pageSize=[Letter](#), char \*\*outmessage=NULL)  
*Member function to open the printer.*
- virtual bool [ClosePrinter](#) (char \*\*outmessage)  
*Close the printer.*
- virtual bool [SetTypeSpacing](#) ([TypeSpacing](#) spacing)  
*Set the the spacing.*
- virtual bool [SetTypeWeight](#) ([TypeWeight](#) weight)  
*Set the type weight.*
- virtual bool [SetTypeSlant](#) ([TypeSlant](#) slant)  
*Set the type slant.*
- virtual bool [NewPage](#) (const string heading="")  
*Perform a page feed and print a heading.*
- virtual bool [PutLine](#) (const string line="")  
*Print out a string and follow it with a new line sequence.*
- virtual bool [Put](#) (const string text)  
*Print a string of text.*
- virtual bool [Tab](#) (int column)  
*Tab over to the specified column.*
- virtual [~PostScriptPrinterDevice](#) ()  
*Destructor.*

### Private Member Functions

- bool [PutPageHeader](#) ()  
*Function to put the page header.*
- const string [PSQuote](#) (const string s) const  
*Function to PostScript quote a string.*

### Private Attributes

- ofstream [printerStream](#)  
*Output stream.*
- string [title](#)  
*The document title.*
- int [pages](#)  
*The page count.*
- int [lines](#)  
*The line count.*
- int [maxLines](#)  
*The maximum number of lines per page.*
- bool [partline](#)  
*Partial line flag.*
- bool [needPageHeader](#)  
*Flag to let us know if we need a page header.*

### Additional Inherited Members

#### 9.121.1 Detailed Description

Derived class for printing on Postscript printers.

Uses a standard 12pt Courier family of fonts and simulates an impact printer.

#### Author

Robert Heller <heller@deepsoft.com>

#### 9.121.2 Constructor & Destructor Documentation

##### 9.121.2.1 PostScriptPrinterDevice()

```
FCFSupport::PostScriptPrinterDevice::PostScriptPrinterDevice (  
    const string filename = "",  
    const string title_ = "",  
    PageSize pageSize = Letter,  
    char ** outmessage = NULL )
```

Constructor.

Create a new printer device instance from a set of parameters, all of which have default values, so this also doubles as the default base constructor.

**Parameters**

<i>filename</i>	Output filename.
<i>title</i>	An internal document title string.
<i>pageSize</i>	The page size to use.
<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur. This parameter is hidden from the Tcl interface.

**9.121.2.2   ~PostScriptPrinterDevice()**

```
virtual FCFSupport::PostScriptPrinterDevice::~~PostScriptPrinterDevice ( ) [virtual]
```

Destructor.

Close the printer.

**9.121.3   Member Function Documentation****9.121.3.1   ClosePrinter()**

```
virtual bool FCFSupport::PostScriptPrinterDevice::ClosePrinter (
    char ** outmessage ) [virtual]
```

Close the printer.

**Parameters**

<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur. This parameter is hidden from the Tcl interface.
-------------------	--

Reimplemented from [FCFSupport::PrinterDevice](#).

**9.121.3.2   NewPage()**

```
virtual bool FCFSupport::PostScriptPrinterDevice::NewPage (
    const string heading = "" ) [virtual]
```

Perform a page feed and print a heading.

## Parameters

<i>heading</i>	The heading string.
----------------	---------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

### 9.121.3.3 OpenPrinter()

```
virtual bool FCFSupport::PostScriptPrinterDevice::OpenPrinter (
    const string filename,
    PageSize pageSize = Letter,
    char ** outmessage = NULL ) [virtual]
```

Member function to open the printer.

## Parameters

<i>filename</i>	Output filename.
<i>pageSize</i>	The page size to use.
<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur. This parameter is hidden from the Tcl interface.

Reimplemented from [FCFSupport::PrinterDevice](#).

### 9.121.3.4 PSQuote()

```
const string FCFSupport::PostScriptPrinterDevice::PSQuote (
    const string s ) const [private]
```

Function to PostScript quote a string.

## Parameters

<i>s</i>	The string to quote.
----------	----------------------

### 9.121.3.5 Put()

```
virtual bool FCFSupport::PostScriptPrinterDevice::Put (
    const string text ) [virtual]
```

Print a string of text.

Don't include a newline.

#### Parameters

<i>text</i>	The string to print.
-------------	----------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

#### 9.121.3.6 PutLine()

```
virtual bool FCFSupport::PostScriptPrinterDevice::PutLine (
    const string line = "" ) [virtual]
```

Print out a string and follow it with a new line sequence.

#### Parameters

<i>line</i>	The line to print.
-------------	--------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

#### 9.121.3.7 PutPageHeader()

```
bool FCFSupport::PostScriptPrinterDevice::PutPageHeader ( ) [private]
```

Function to put the page header.

#### 9.121.3.8 SetTypeSlant()

```
virtual bool FCFSupport::PostScriptPrinterDevice::SetTypeSlant (
    TypeSlant slant ) [virtual]
```

Set the type slant.

#### Parameters

<i>slant</i>	The new type slant.
--------------	---------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

#### 9.121.3.9 SetTypeSpacing()

```
virtual bool FCFSupport::PostScriptPrinterDevice::SetTypeSpacing (
    TypeSpacing spacing ) [virtual]
```

Set the the spacing.

##### Parameters

<i>spacing</i>	The new type spacing.
----------------	-----------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

#### 9.121.3.10 SetTypeWeight()

```
virtual bool FCFSupport::PostScriptPrinterDevice::SetTypeWeight (
    TypeWeight weight ) [virtual]
```

Set the type weight.

##### Parameters

<i>weight</i>	The new type weight.
---------------	----------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

#### 9.121.3.11 Tab()

```
virtual bool FCFSupport::PostScriptPrinterDevice::Tab (
    int column ) [virtual]
```

Tab over to the specified column.

##### Parameters

<i>column</i>	The desired tab column.
---------------	-------------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

## 9.121.4 Member Data Documentation

### 9.121.4.1 lines

```
int FCFSupport::PostScriptPrinterDevice::lines [private]
```

The line count.

### 9.121.4.2 maxLines

```
int FCFSupport::PostScriptPrinterDevice::maxLines [private]
```

The maximum number of lines per page.

### 9.121.4.3 needPageHeader

```
bool FCFSupport::PostScriptPrinterDevice::needPageHeader [private]
```

Flag to let us know if we need a page header,.

### 9.121.4.4 pages

```
int FCFSupport::PostScriptPrinterDevice::pages [private]
```

The page count.

### 9.121.4.5 partline

```
bool FCFSupport::PostScriptPrinterDevice::partline [private]
```

Partial line flag.



## 9.121.4.6 printerStream

```
ofstream FCFSupport::PostScriptPrinterDevice::printerStream [private]
```

Output stream.

## 9.121.4.7 title

```
string FCFSupport::PostScriptPrinterDevice::title [private]
```

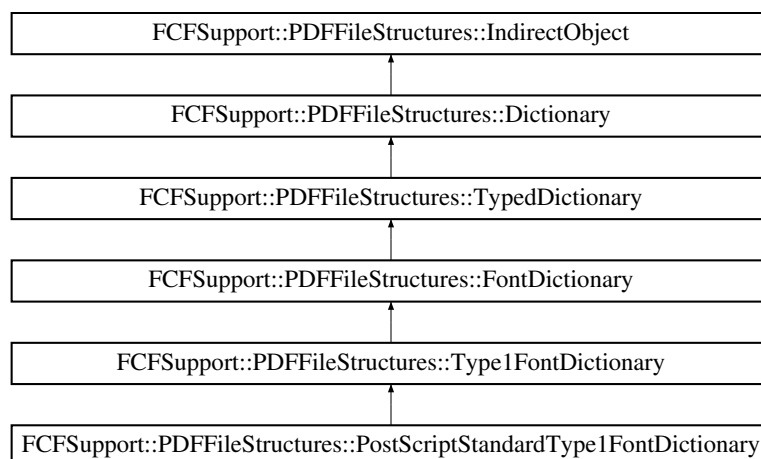
The document title.

## 9.122 FCFSupport::PDFFileStructures::PostScriptStandardType1FontDictionary Class Reference

A standard Type1 PostScript font dictionary.

```
#include <PDFPrinterSupport.h>
```

Inheritance diagram for FCFSupport::PDFFileStructures::PostScriptStandardType1FontDictionary:



### Public Member Functions

- [PostScriptStandardType1FontDictionary](#) (const string name, unsigned long int objNum=0L, unsigned short int genNum=0, CrossReferenceTable \*tab=NULL)

*Constructor.*

- [~PostScriptStandardType1FontDictionary](#) ()

*Destructor.*

## Additional Inherited Members

### 9.122.1 Detailed Description

A standard Type1 PostScript font dictionary.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.122.2 Constructor & Destructor Documentation

#### 9.122.2.1 PostScriptStandardType1FontDictionary()

```
FCFSupport::PDFFileStructures::PostScriptStandardType1FontDictionary::PostScriptStandardType1FontDictionary (
    const string name,
    unsigned long int objNum = 0L,
    unsigned short int genNum = 0,
    CrossReferenceTable * tab = NULL ) [inline]
```

Constructor.

Construct one of the 14 standard PostScript fonts.

#### Parameters

<i>name</i>	The name of the PostScript font.
<i>objNum</i>	The next object number.
<i>genNum</i>	The generation number.
<i>tab</i>	The cross reference table we are in.

#### 9.122.2.2 ~PostScriptStandardType1FontDictionary()

```
FCFSupport::PDFFileStructures::PostScriptStandardType1FontDictionary::~~PostScriptStandardType1FontDictionary ( ) [inline]
```

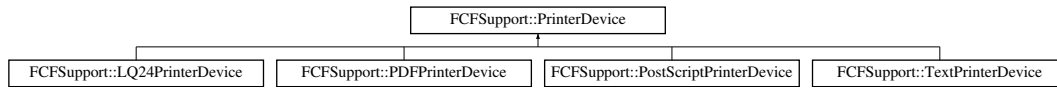
Destructor.

## 9.123 FCFSupport::PrinterDevice Class Reference

Base class for printer devices (hard copy output).

```
#include <Printer.h>
```

Inheritance diagram for FCFSupport::PrinterDevice:



### Public Types

- enum [PageSize](#) { [Letter](#), [A4](#) }  
*Page size selection, for those printers that support different page sizes.*
- enum [TypeSpacing](#) { [One](#), [Half](#), [Double](#) }  
*Horizontal type spacing.*
- enum [TypeWeight](#) { [Normal](#), [Bold](#) }  
*Type weight.*
- enum [TypeSlant](#) { [Roman](#), [Italic](#) }  
*Type slant.*

### Public Member Functions

- [PrinterDevice](#) (const string filename="", const string title="", [PageSize](#) pageSize\_[\\_Letter](#), char \*\*outmessage=NULL)
- *Constructor.*
- virtual bool [OpenPrinter](#) (const string filename, [PageSize](#) pageSize\_[\\_Letter](#), char \*\*outmessage=NULL)
- *Member function to open the printer.*
- virtual bool [ClosePrinter](#) (char \*\*outmessage)
- *Close the printer.*
- bool [IsOpenP](#) () const
- *Is the printer open?*
- [PageSize](#) [PrinterPageSize](#) () const
- *Return the page size.*
- virtual bool [SetTypeSpacing](#) ([TypeSpacing](#) spacing)
- *Set the the spacing.*
- virtual bool [SetTypeWeight](#) ([TypeWeight](#) weight)
- *Set the type weight.*
- virtual bool [SetTypeSlant](#) ([TypeSlant](#) slant)
- *Set the type slant.*
- virtual bool [NewPage](#) (const string heading="")
- *Perform a page feed and print a heading.*
- virtual bool [PutLine](#) (const string line="")
- *Print out a string and follow it with a new line sequence.*

- virtual bool `Put` (const string text)  
*Print a string of text.*
- virtual bool `Put` (int number)  
*Print an integer.*
- virtual bool `Put` (double number)  
*Print a double.*
- virtual bool `Tab` (int column)  
*Tab over to the specified column.*
- virtual `~PrinterDevice` ()  
*Destructor.*

## Protected Attributes

- bool `isOpenP`  
*Is open flag.*
- `PageSize` `pageSize`  
*Document page size.*

### 9.123.1 Detailed Description

Base class for printer devices (hard copy output).

Defines a very basic set of printing operations, including printing strings, numbers, lines, form feeds, tabbing, and changing the spacing, weight, and slant of the type used.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.123.2 Member Enumeration Documentation

#### 9.123.2.1 PageSize

```
enum FCFSupport::PrinterDevice::PageSize
```

Page size selection, for those printers that support different page sizes.

#### Enumerator

Letter	US Letter page size.
A4	European A4 page size.

### 9.123.2.2 TypeSlant

```
enum FCFSupport::PrinterDevice::TypeSlant
```

Type slant.

Enumerator

Roman	Upright.
Italic	Italic.

### 9.123.2.3 TypeSpacing

```
enum FCFSupport::PrinterDevice::TypeSpacing
```

Horizontal type spacing.

This is the character width.

Enumerator

One	Single wide characters. Normal width charactes.
Half	Half (actually 60%) wide characters. Condensed printing.
Double	Double wide characters.

### 9.123.2.4 TypeWeight

```
enum FCFSupport::PrinterDevice::TypeWeight
```

Type weight.

Enumerator

Normal	Normal weight.
Bold	Heavy (bold) weight.

### 9.123.3 Constructor & Destructor Documentation

#### 9.123.3.1 PrinterDevice()

```
FCFSupport::PrinterDevice::PrinterDevice (
    const string filename = "",
    const string title = "",
    PageSize pageSize_ = Letter,
    char ** outmessage = NULL ) [inline]
```

Constructor.

Create a new printer device instance from a set of parameters, all of which have default values, so this also doubles as the default base constructor.

##### Parameters

<i>filename</i>	Output filename.
<i>title</i>	An internal document title string.
<i>pageSize_</i>	The page size to use.
<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur. This parameter is hidden from the Tcl interface.

References `isOpenP`, `Letter`, and `OpenPrinter()`.

#### 9.123.3.2 ~PrinterDevice()

```
virtual FCFSupport::PrinterDevice::~~PrinterDevice ( ) [inline], [virtual]
```

Destructor.

Close the printer.

References `ClosePrinter()`.

### 9.123.4 Member Function Documentation

#### 9.123.4.1 ClosePrinter()

```
virtual bool FCFSupport::PrinterDevice::ClosePrinter (
    char ** outmessage ) [inline], [virtual]
```

Close the printer.

## Parameters

<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur. This parameter is hidden from the Tcl interface.
-------------------	--

Reimplemented in [FCFSupport::TextPrinterDevice](#), [FCFSupport::LQ24PrinterDevice](#), [FCFSupport::PostScriptPrinterDevice](#), and [FCFSupport::PDFPrinterDevice](#).

References [isOpenP](#).

Referenced by [~PrinterDevice\(\)](#).

9.123.4.2 [isOpenP\(\)](#)

```
bool FCFSupport::PrinterDevice::isOpenP ( ) const [inline]
```

Is the printer open?

References [isOpenP](#).

9.123.4.3 [NewPage\(\)](#)

```
virtual bool FCFSupport::PrinterDevice::NewPage (
    const string heading = "" ) [inline], [virtual]
```

Perform a page feed and print a heading.

## Parameters

<i>heading</i>	The heading string.
----------------	---------------------

Reimplemented in [FCFSupport::LQ24PrinterDevice](#), [FCFSupport::PostScriptPrinterDevice](#), [FCFSupport::PDFPrinterDevice](#), and [FCFSupport::TextPrinterDevice](#).

References [PutLine\(\)](#).

Referenced by [SetTypeSlant\(\)](#).

#### 9.123.4.4 OpenPrinter()

```
virtual bool FCFSupport::PrinterDevice::OpenPrinter (
    const string filename,
    PageSize pageSize_ = Letter,
    char ** outmessage = NULL ) [inline], [virtual]
```

Member function to open the printer.

##### Parameters

<i>filename</i>	Output filename.
<i>pageSize</i>	The page size to use.
<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur. This parameter is hidden from the Tcl interface.

Reimplemented in [FCFSupport::TextPrinterDevice](#), [FCFSupport::PostScriptPrinterDevice](#), [FCFSupport::LQ24PrinterDevice](#), and [FCFSupport::PDFPrinterDevice](#).

References `isOpenP`.

Referenced by `PrinterDevice()`.

#### 9.123.4.5 PrinterPageSize()

```
PageSize FCFSupport::PrinterDevice::PrinterPageSize ( ) const [inline]
```

Return the page size.

References `pageSize`.

#### 9.123.4.6 Put() [1/3]

```
virtual bool FCFSupport::PrinterDevice::Put (
    const string text ) [inline], [virtual]
```

Print a string of text.

Don't include a newline.

##### Parameters

<i>text</i>	The string to print.
-------------	----------------------



Reimplemented in [FCFSupport::LQ24PrinterDevice](#), [FCFSupport::PostScriptPrinterDevice](#), [FCFSupport::PDFPrinterDevice](#), and [FCFSupport::TextPrinterDevice](#).

Referenced by [Put\(\)](#), and [PutLine\(\)](#).

#### 9.123.4.7 Put() [2/3]

```
virtual bool FCFSupport::PrinterDevice::Put (  
    int number ) [inline], [virtual]
```

Print an integer.

Don't include a newline.

##### Parameters

<i>number</i>	The string to print.
---------------	----------------------

References [Put\(\)](#).

#### 9.123.4.8 Put() [3/3]

```
virtual bool FCFSupport::PrinterDevice::Put (  
    double number ) [inline], [virtual]
```

Print a double.

Don't include a newline.

##### Parameters

<i>number</i>	The string to print.
---------------	----------------------

References [Put\(\)](#).

#### 9.123.4.9 PutLine()

```
virtual bool FCFSupport::PrinterDevice::PutLine (  
    const string line = "" ) [inline], [virtual]
```

Print out a string and follow it with a new line sequence.

## Parameters

<i>line</i>	The line to print.
-------------	--------------------

Reimplemented in [FCFSupport::LQ24PrinterDevice](#), [FCFSupport::PostScriptPrinterDevice](#), [FCFSupport::PDFPrinterDevice](#), and [FCFSupport::TextPrinterDevice](#).

References Put().

Referenced by NewPage().

#### 9.123.4.10 SetTypeSlant()

```
virtual bool FCFSupport::PrinterDevice::SetTypeSlant (
    TypeSlant slant ) [inline], [virtual]
```

Set the type slant.

## Parameters

<i>slant</i>	The new type slant.
--------------	---------------------

Reimplemented in [FCFSupport::LQ24PrinterDevice](#), [FCFSupport::PostScriptPrinterDevice](#), and [FCFSupport::PDFPrinterDevice](#).

References NewPage().

#### 9.123.4.11 SetTypeSpacing()

```
virtual bool FCFSupport::PrinterDevice::SetTypeSpacing (
    TypeSpacing spacing ) [inline], [virtual]
```

Set the the spacing.

## Parameters

<i>spacing</i>	The new type spacing.
----------------	-----------------------

Reimplemented in [FCFSupport::LQ24PrinterDevice](#), [FCFSupport::PostScriptPrinterDevice](#), and [FCFSupport::PDFPrinterDevice](#).

#### 9.123.4.12 SetTypeWeight()

```
virtual bool FCFSupport::PrinterDevice::SetTypeWeight (
    TypeWeight weight ) [inline], [virtual]
```

Set the type weight.

##### Parameters

<i>weight</i>	The new type weight.
---------------	----------------------

Reimplemented in [FCFSupport::LQ24PrinterDevice](#), [FCFSupport::PostScriptPrinterDevice](#), and [FCFSupport::PDFPrinterDevice](#).

#### 9.123.4.13 Tab()

```
virtual bool FCFSupport::PrinterDevice::Tab (
    int column ) [inline], [virtual]
```

Tab over to the specified column.

##### Parameters

<i>column</i>	The desired tab column.
---------------	-------------------------

Reimplemented in [FCFSupport::LQ24PrinterDevice](#), [FCFSupport::PostScriptPrinterDevice](#), [FCFSupport::PDFPrinterDevice](#), and [FCFSupport::TextPrinterDevice](#).

### 9.123.5 Member Data Documentation

#### 9.123.5.1 isOpenP

```
bool FCFSupport::PrinterDevice::isOpenP [protected]
```

Is open flag.

Referenced by [ClosePrinter\(\)](#), [IsOpenP\(\)](#), [OpenPrinter\(\)](#), and [PrinterDevice\(\)](#).

### 9.123.5.2 pageSize

`PageSize` `FCFSupport::PrinterDevice::pageSize` [protected]

Document page size.

Referenced by `PrinterPageSize()`.

## 9.124 CTCPanel::PushButton Class Reference

Push Button object type.

### Public Member Functions

- `PushButton` (name, \_ctcpanel, \_canvas,...)  
*Construct a `PushButton` object.*
- `~PushButton` ()  
*Clean up all data objects and free up all resources.*
- `getv` ()  
*Method to get our value (`PushButton` state).*
- `setv` (newstate)  
*Method to set out value (`PushButton` state).*
- `geti` (ind)  
*Method to get the state of one of our indicators.*
- `seti` (ind, value)  
*Method to set an indicator's state (none).*
- `invoke` ()  
*Method to invoke the `PushButton`.*

### Private Member Functions

- `_configureLabel` (option, value)  
*Method to update the label option.*

### Private Attributes

- `ctcpanel`  
*The CTC Panel component (parent widget).*
- `canvas`  
*The canvas component (parent widget component).*

### 9.124.1 Detailed Description

Push Button object type.

These are on the control panel and represent simple push buttons.

## Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The control panel canvas to draw the <a href="#">PushButton</a> on.
<code>...</code>	Options: <ul style="list-style-type: none"> <li>• <code>-x</code> The x coordinate of the object (readonly, default 0).</li> <li>• <code>-y</code> The y coordinate of the object (readonly, default 0).</li> <li>• <code>-controlpoint</code> The name of the control point this <a href="#">PushButton</a> is part of (readonly, default CP1).</li> <li>• <code>-color</code> The color of the <a href="#">PushButton</a> (default white).</li> <li>• <code>-label</code> The label of the <a href="#">PushButton</a> (default "PushButton").</li> <li>• <code>-command</code> The Tcl script to run when the <a href="#">PushButton</a> button is pushed.</li> </ul>

Defined coords terminals: none. Defined values (states): none. Defined indicators:

- `-color` if on.

## Author

Robert Heller <heller@deepsoft.com>

## 9.124.2 Constructor &amp; Destructor Documentation

## 9.124.2.1 PushButton()

```
CTCPanel::PushButton::PushButton (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a [PushButton](#) object.

## Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The control panel canvas to draw the <a href="#">PushButton</a> on.
<code>...</code>	Option list.

#### 9.124.2.2 ~PushButton()

```
CTCPanel::PushButton::~~PushButton ( )
```

Clean up all data objects and free up all resources.

### 9.124.3 Member Function Documentation

#### 9.124.3.1 \_configureLabel()

```
CTCPanel::PushButton::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

#### 9.124.3.2 geti()

```
CTCPanel::PushButton::geti (
    ind )
```

Method to get the state of one of our indicators.

#### 9.124.3.3 getv()

```
CTCPanel::PushButton::getv ( )
```

Method to get our value ([PushButton](#) state).

#### 9.124.3.4 invoke()

```
CTCPanel::PushButton::invoke ( )
```

Method to invoke the [PushButton](#).

#### 9.124.3.5 seti()

```
CTCPanel::PushButton::seti (
    ind ,
    value )
```

Method to set an indicator's state (none).

#### 9.124.3.6 setv()

```
CTCPanel::PushButton::setv (
    newstate )
```

Method to set out value ([PushButton](#) state).

##### Parameters

<i>newstate</i>	The new <a href="#">PushButton</a> state.
-----------------	---

### 9.124.4 Member Data Documentation

#### 9.124.4.1 canvas

```
CTCPanel::PushButton::canvas [private]
```

The canvas component (parent widget component).

#### 9.124.4.2 ctcpnl

```
CTCPanel::PushButton::ctcpnl [private]
```

The CTC Panel component (parent widget).

## 9.125 raildriver::RaildriverClient Class Reference

Raildriver Client class – connects to the Raildriver daemon.

## Public Member Functions

- [RaildriverClient](#) (name,...)  
*Construct a [RaildriverClient](#) object.*
- [~RaildriverClient](#) ()  
*close the connection.*
- [clear](#) ()  
*Send a CLEAR message to the daemon.*
- [mask](#) (...)  
*Send a MASK message to the daemon.*
- [leds](#) (ledstring)  
*Send a LED message to the daemon.*
- [speaker](#) (onoff)  
*Turn the speaker on or off.*

## Private Member Functions

- [\\_readevent](#) ()  
*Handle messages from the daemon.*
- [\\_poller](#) ()  
*Polling function.*

## Private Attributes

- [socket](#)  
*The socket descriptor connected to the daemon.*
- [pollid](#)  
*Holds the poll after id.*

### 9.125.1 Detailed Description

Raildriver Client class – connects to the Raildriver daemon.

Polls at intervals for Raildriver input events.

Options:

- -port Port (on localhost) to connect to. The default is 41000.
- -pollinterval Interval in milliseconds (between 250 and 2000) to poll the daemon. The default is 500.
- -pollevents List of events to poll for. See [raildriver::RaildriverEvents](#) for the allowed element values. The default is the empty list.
- -eventhandler A script (at the global level) to evaluate when a message arrives from the daemon. Two elements are appended: the message status code and the text of the message. The default is no handler.



## 9.125.2 Constructor & Destructor Documentation

### 9.125.2.1 RaildriverClient()

```
raildriver::RaildriverClient::RaildriverClient (
    name ,
    ... )
```

Construct a [RaildriverClient](#) object.

### 9.125.2.2 ~RaildriverClient()

```
raildriver::RaildriverClient::~~RaildriverClient ( )
```

close the connection.

## 9.125.3 Member Function Documentation

### 9.125.3.1 \_poller()

```
raildriver::RaildriverClient::_poller ( ) [private]
```

Polling function.

### 9.125.3.2 \_readevent()

```
raildriver::RaildriverClient::_readevent ( ) [private]
```

Handle messages from the daemon.

### 9.125.3.3 clear()

```
raildriver::RaildriverClient::clear ( )
```

Send a CLEAR message to the daemon.

### 9.125.3.4 leds()

```
raildriver::RaildriverClient::leds (
    ledstring )
```

Send a LED message to the daemon.

**Parameters**

<i>ledstring</i>	Led string to display.
------------------	------------------------

**9.125.3.5 mask()**

```
raildriver::RaildriverClient::mask (  
    ... )
```

Send a MASK message to the daemon.

**Parameters**

...	Mask values
-----	-------------

**9.125.3.6 speaker()**

```
raildriver::RaildriverClient::speaker (  
    onoff )
```

Turn the speaker on or off.

**Parameters**

<i>onoff</i>	Boolean indicating on (true) or off (false).
--------------	--

**9.125.4 Member Data Documentation****9.125.4.1 pollid**

```
raildriver::RaildriverClient::pollid [private]
```

Holds the poll after id.

## 9.125.4.2 socket

```
raildriver::RaildriverClient::socket [private]
```

The socket descriptor connected to the daemon.

## 9.126 RaildriverIO Class Reference

Low-level Raildriver I/O functions.

```
#include <RaildriverIO.h>
```

### Public Types

- enum [Eventmask\\_bits](#) {  
[NONE\\_M](#) = 0, [REVERSER\\_M](#) = 1 << 0, [THROTTLE\\_M](#) = 1 << 1, [AUTOBRAKE\\_M](#) = 1 << 2,  
[INDEPENDBRK\\_M](#) = 1 << 3, [BAILOFF\\_M](#) = 1 << 4, [WIPER\\_M](#) = 1 << 5, [HEADLIGHT\\_M](#) = 1 << 6,  
[DIGITAL1\\_M](#) = 1 << 7, [DIGITAL2\\_M](#) = 1 << 8, [DIGITAL3\\_M](#) = 1 << 9, [DIGITAL4\\_M](#) = 1 << 10,  
[DIGITAL5\\_M](#) = 1 << 11, [DIGITAL6\\_M](#) = 1 << 12 }

*Event Masks.*

- enum [Eventcodes](#) {  
[NONE](#) = 0, [REVERSER](#), [THROTTLE](#), [AUTOBRAKE](#),  
[INDEPENDBRK](#), [BAILOFF](#), [WIPER](#), [HEADLIGHT](#),  
[DIGITAL1](#), [DIGITAL2](#), [DIGITAL3](#), [DIGITAL4](#),  
[DIGITAL5](#), [DIGITAL6](#) }

*Event Codes.*

### Public Member Functions

- [RaildriverIO](#) (const char \*path, char \*\*outmessage=NULL)  
*Constructor.*
- [~RaildriverIO](#) ()  
*Destructor.*
- void [SetLEDS](#) (const char \*ledstring, char \*\*outmessage=NULL)  
*Set the Speedometer LEDs.*
- void [SpeakerOn](#) (char \*\*outmessage=NULL)  
*Turn the speaker on.*
- void [SpeakerOff](#) (char \*\*outmessage=NULL)  
*Turn the speaker off.*
- unsigned char [GetReverser](#) () const  
*Get Reverser value (0-255).*
- unsigned char [GetThrottle](#) () const  
*Get Throttle value (0-255).*
- unsigned char [GetAutoBrake](#) () const  
*Get Auto Brake value (0-255).*

- unsigned char [GetIndependBrake](#) () const  
*Get Independenent Brake value (0-255).*
- unsigned char [GetBailOff](#) () const  
*Get Bail Off value (0-255).*
- unsigned char [GetHeadlight](#) () const  
*Get Headlight value (0-255).*
- unsigned char [GetWiper](#) () const  
*Get Wiper value (0-255).*
- bool [GetBlueButton1](#) () const  
*Get Blue Button 1.*
- bool [GetBlueButton2](#) () const  
*Get Blue Button 2.*
- bool [GetBlueButton3](#) () const  
*Get Blue Button 3.*
- bool [GetBlueButton4](#) () const  
*Get Blue Button 4.*
- bool [GetBlueButton5](#) () const  
*Get Blue Button 5.*
- bool [GetBlueButton6](#) () const  
*Get Blue Button 6.*
- bool [GetBlueButton7](#) () const  
*Get Blue Button 7.*
- bool [GetBlueButton8](#) () const  
*Get Blue Button 8.*
- bool [GetBlueButton9](#) () const  
*Get Blue Button 9.*
- bool [GetBlueButton10](#) () const  
*Get Blue Button 10.*
- bool [GetBlueButton11](#) () const  
*Get Blue Button 11.*
- bool [GetBlueButton12](#) () const  
*Get Blue Button 12.*
- bool [GetBlueButton13](#) () const  
*Get Blue Button 13.*
- bool [GetBlueButton14](#) () const  
*Get Blue Button 14.*
- bool [GetBlueButton15](#) () const  
*Get Blue Button 15.*
- bool [GetBlueButton16](#) () const  
*Get Blue Button 16.*
- bool [GetBlueButton17](#) () const  
*Get Blue Button 17.*
- bool [GetBlueButton18](#) () const  
*Get Blue Button 18.*
- bool [GetBlueButton19](#) () const  
*Get Blue Button 19.*
- bool [GetBlueButton20](#) () const

- Get Blue Button 20.*
- bool [GetBlueButton21](#) () const
- Get Blue Button 21.*
- bool [GetBlueButton22](#) () const
- Get Blue Button 22.*
- bool [GetBlueButton23](#) () const
- Get Blue Button 23.*
- bool [GetBlueButton24](#) () const
- Get Blue Button 24.*
- bool [GetBlueButton25](#) () const
- Get Blue Button 25.*
- bool [GetBlueButton26](#) () const
- Get Blue Button 26.*
- bool [GetBlueButton27](#) () const
- Get Blue Button 27.*
- bool [GetBlueButton28](#) () const
- Get Blue Button 28.*
- bool [GetZoomUp](#) () const
- Get Zoom Up.*
- bool [GetZoopDown](#) () const
- Get Zoom Down.*
- bool [GetPanUp](#) () const
- Get Pan Up.*
- bool [GetPanRight](#) () const
- Get Pan Right.*
- bool [GetPanDown](#) () const
- Get Pan Down.*
- bool [GetPanLeft](#) () const
- Get Pan Left.*
- bool [GetRangeUp](#) () const
- Get Range Up.*
- bool [GetRangeDown](#) () const
- Get Range Down.*
- bool [GetEBrakeUp](#) () const
- Get Emergency Brake Up.*
- bool [GetEBrakeDown](#) () const
- Get Emergency Brake Down.*
- bool [GetAlert](#) () const
- Get Alert.*
- bool [GetSand](#) () const
- Get Sand.*
- bool [GetPantograph](#) () const
- Get Pantograph.*
- bool [GetBell](#) () const
- Get Bell.*
- bool [GetWhistleUp](#) () const
- Get Whistle Up.*

- bool [GetWhistleDown](#) () const  
*Get Whistle Down.*
- unsigned char [GetProductCodeId](#) () const  
*Get Product Code Id.*
- bool [ReadInputs](#) ([Eventmask\\_bits](#) &mask, int &status)  
*Poll the interface.*

## Private Member Functions

- [RaildriverIO](#) ()  
*Default constructor.*

## Private Attributes

- hid\_device \* [rdriverdev](#)  
*Rail Driver Device.*
- union {  
  unsigned char [ReportBuffer](#) [14]  
    *Event Buffer.*  
  struct **bytes** {  
    unsigned char [Reverser](#)  
      *Reverser lever, 0-255.*  
    unsigned char [Throttle](#)  
      *Throttle / Dynamic Brake lever, 0-255.*  
    unsigned char [AutoBrake](#)  
      *Automatic Brake lever, 0-255.*  
    unsigned char [IndependBrake](#)  
      *Independent Brake lever, 0-255.*  
    unsigned char [BailOff](#)  
      *Bail Off (Independent Brake lever), 0-255.*  
    unsigned char [Wiper](#)  
      *Wiper switch, 0-255.*  
    unsigned char [Headlight](#)  
      *Headlight switch, 0-255.*  
    unsigned char [Digital1](#)  
      *Blue Buttons 1-8.*  
    unsigned char [Digital2](#)  
      *Blue Buttons 9-16.*  
    unsigned char [Digital3](#)  
      *Blue Buttons 17-24.*  
    unsigned char [Digital4](#)  
      *Blue Buttons 25-28, Zoom, Pan Buttons.*  
    unsigned char [Digital5](#)  
      *Pan, Cab buttons.*  
    unsigned char [Digital6](#)  
      *Cab Buttons, Whistle Switch.*  
    unsigned char [ProductCodeId](#)  
      *Product Code Id, usually 210.*  
  } [theBytes](#)  
} [RDInput](#)  
  
*Event data.*

## Static Private Attributes

- static const unsigned short int [PIEngineering](#)  
*Rail Driver vendor code.*
- static const unsigned short int [RailDriverModernDesktop](#)  
*Rail Driver product code.*
- static const int [LEDCommand](#)  
*LED Command code.*
- static const int [SpeakerCommand](#)  
*Speaker command code.*

### 9.126.1 Detailed Description

Low-level Raildriver I/O functions.

This class implements the low-level Raildriver I/O functions and provides access to the two endpoints, one input (controls, switches, and buttons) and one output (LED display and speaker switch).

The Raildriver "game" console contains a collection of levers, buttons, and switches that simulate a locomotive control stand. There is a reverser lever, a throttle, two brake levers, switches for the lights and wipers, and buttons, switches, and levers for things like the bell, alerter, whistle (or horn), sand, pantograph, and other functions, plus a collection of general purpose buttons that can be programmed to provide any other function. The console also contains a three digit seven-segment display and there is a built in speaker.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.126.2 Tcl Package Provided

Raildriverio 1.0.0

### 9.126.3 Library Provided

libraildriverio 1.0.0

### 9.126.4 Member Enumeration Documentation

#### 9.126.4.1 Eventcodes

```
enum RaildriverIO::Eventcodes
```

Event Codes.

These are the event codes for the Rail Driver's report message. There is a code for each of the thirteen bytes in the report buffer.

## Enumerator

NONE	No bits set.
REVERSER	Reverser lever. This is a value between 0 and 255 representing the position of the reverser lever.
THROTTLE	Throttle lever. This is a value between 0 and 255 representing the position of the throttle / dynamic brake lever.
AUTOBRAKE	Automatic Brake lever. This is a value between 0 and 255 representing the position of the automatic brake lever.
INDEPENDBRK	Independent Brake lever. This is a value between 0 and 255 representing the position of the independent brake lever.
BAILOFF	Independent Brake bail off. This is a value between 0 and 255 representing the position of the independent brake lever bail off.
WIPER	Wiper switch. This is a value between 0 and 255 representing the position of the wiper switch.
HEADLIGHT	Headlight switch. This is a value between 0 and 255 representing the position of the headlight switch.
DIGITAL1	Blue Buttons 1-8. This is a bitfield representing 8 of the generic "blue" buttons.
DIGITAL2	Blue Buttons 9-16. This is a bitfield representing 8 of the generic "blue" buttons.
DIGITAL3	Blue Buttons 17-24. This is a bitfield representing 8 of the generic "blue" buttons.
DIGITAL4	Blue Buttons 25-28, Zoom, Pan. This is a bitfield representing the last 4 of the generic "blue" buttons, the zoom rocker, and one-half of the pan (2d) rocker.
DIGITAL5	Pan, Cab Buttons. This is a bitfield representing the second half of the pan (2d) rocker, and several of the two of the cab rocker switches.
DIGITAL6	Cab Buttons, Whistle. This is a bitfield representing the cab buttons and the whistle lever.

## 9.126.4.2 Eventmask\_bits

```
enum RaildriverIO::Eventmask_bits
```

## Event Masks.

These are the mask bits for the Rail Driver's report message. Each bit represents one of the thirteen bytes in the report buffer.

## Enumerator

NONE_M	No bits set. This is the empty mask.
REVERSER_M	Reverser lever. This is a value between 0 and 255 representing the position of the reverser lever.
THROTTLE_M	Throttle lever. This is a value between 0 and 255 representing the position of the throttle / dynamic brake lever.
AUTOBRAKE_M	Automatic Brake lever. This is a value between 0 and 255 representing the position of the automatic brake lever.
INDEPENDBRK_M	Independent Brake lever. This is a value between 0 and 255 representing the position of the independent brake lever.
BAILOFF_M	Independent Brake bail off. This is a value between 0 and 255 representing the position of the independent brake lever bail off.



## Enumerator

WIPER_M	Wiper switch. This is a value between 0 and 255 representing the position of the wiper switch.
HEADLIGHT_M	Headlight switch. This is a value between 0 and 255 representing the position of the headlight switch.
DIGITAL1_M	Blue Buttons 1-8. This is a bitfield representing 8 of the generic "blue" buttons.
DIGITAL2_M	Blue Buttons 9-16. This is a bitfield representing 8 of the generic "blue" buttons.
DIGITAL3_M	Blue Buttons 17-24. This is a bitfield representing 8 of the generic "blue" buttons.
DIGITAL4_M	Blue Buttons 25-28, Zoom, Pan. This is a bitfield representing the last 4 of the generic "blue" buttons, the zoom rocker, and one-half of the pan (2d) rocker.
DIGITAL5_M	Pan, Cab Buttons. This is a bitfield representing the second half of the pan (2d) rocker, and several of the two of the cab rocker switches.
DIGITAL6_M	Cab Buttons, whistle. This is a bitfield representing the cab buttons and the whistle lever.

## 9.126.5 Constructor &amp; Destructor Documentation

## 9.126.5.1 RaildriverIO() [1/2]

```
RaildriverIO::RaildriverIO (
    const char * path,
    char ** outmessage = NULL )
```

Constructor.

The argument is the bus number and device number that identifies the specific device. Finds and opens the device and initializes various data objects, generally preparing for I/O to the connected rail driver console.

## Parameters

<i>path</i>	The path name or the empty string or NULL.
<i>outmessage</i>	Receives error messages.

## 9.126.5.2 ~RaildriverIO()

```
RaildriverIO::~~RaildriverIO ( )
```

Destructor.

Closes the device and free up system resources.

### 9.126.5.3 RaildriverIO() [2/2]

```
RaildriverIO::RaildriverIO ( ) [inline], [private]
```

Default constructor.

This constructor is never called. It is made private to force a compiler error if an attempt is made to use it.

References PIEngineering.

## 9.126.6 Member Function Documentation

### 9.126.6.1 GetAlert()

```
bool RaildriverIO::GetAlert ( ) const [inline]
```

Get Alert.

References RDInput.

### 9.126.6.2 GetAutoBrake()

```
unsigned char RaildriverIO::GetAutoBrake ( ) const [inline]
```

Get Auto Brake value (0-255).

References RDInput.

### 9.126.6.3 GetBailOff()

```
unsigned char RaildriverIO::GetBailOff ( ) const [inline]
```

Get Bail Off value (0-255).

References RDInput.

#### 9.126.6.4 GetBell()

```
bool RaildriverIO::GetBell ( ) const [inline]
```

Get Bell.

References RDInput.

#### 9.126.6.5 GetBlueButton1()

```
bool RaildriverIO::GetBlueButton1 ( ) const [inline]
```

Get Blue Button 1.

References RDInput.

#### 9.126.6.6 GetBlueButton10()

```
bool RaildriverIO::GetBlueButton10 ( ) const [inline]
```

Get Blue Button 10.

References RDInput.

#### 9.126.6.7 GetBlueButton11()

```
bool RaildriverIO::GetBlueButton11 ( ) const [inline]
```

Get Blue Button 11.

References RDInput.

#### 9.126.6.8 GetBlueButton12()

```
bool RaildriverIO::GetBlueButton12 ( ) const [inline]
```

Get Blue Button 12.

References RDInput.

**9.126.6.9   GetBlueButton13()**

```
bool RaildriverIO::GetBlueButton13 ( ) const [inline]
```

Get Blue Button 13.

References RDInput.

**9.126.6.10   GetBlueButton14()**

```
bool RaildriverIO::GetBlueButton14 ( ) const [inline]
```

Get Blue Button 14.

References RDInput.

**9.126.6.11   GetBlueButton15()**

```
bool RaildriverIO::GetBlueButton15 ( ) const [inline]
```

Get Blue Button 15.

References RDInput.

**9.126.6.12   GetBlueButton16()**

```
bool RaildriverIO::GetBlueButton16 ( ) const [inline]
```

Get Blue Button 16.

References RDInput.

**9.126.6.13   GetBlueButton17()**

```
bool RaildriverIO::GetBlueButton17 ( ) const [inline]
```

Get Blue Button 17.

References RDInput.

**9.126.6.14 GetBlueButton18()**

```
bool RaildriverIO::GetBlueButton18 ( ) const [inline]
```

Get Blue Button 18.

References RDInput.

**9.126.6.15 GetBlueButton19()**

```
bool RaildriverIO::GetBlueButton19 ( ) const [inline]
```

Get Blue Button 19.

References RDInput.

**9.126.6.16 GetBlueButton2()**

```
bool RaildriverIO::GetBlueButton2 ( ) const [inline]
```

Get Blue Button 2.

References RDInput.

**9.126.6.17 GetBlueButton20()**

```
bool RaildriverIO::GetBlueButton20 ( ) const [inline]
```

Get Blue Button 20.

References RDInput.

**9.126.6.18 GetBlueButton21()**

```
bool RaildriverIO::GetBlueButton21 ( ) const [inline]
```

Get Blue Button 21.

References RDInput.

**9.126.6.19 GetBlueButton22()**

```
bool RaildriverIO::GetBlueButton22 ( ) const [inline]
```

Get Blue Button 22.

References RDInput.

**9.126.6.20 GetBlueButton23()**

```
bool RaildriverIO::GetBlueButton23 ( ) const [inline]
```

Get Blue Button 23.

References RDInput.

**9.126.6.21 GetBlueButton24()**

```
bool RaildriverIO::GetBlueButton24 ( ) const [inline]
```

Get Blue Button 24.

References RDInput.

**9.126.6.22 GetBlueButton25()**

```
bool RaildriverIO::GetBlueButton25 ( ) const [inline]
```

Get Blue Button 25.

References RDInput.

**9.126.6.23 GetBlueButton26()**

```
bool RaildriverIO::GetBlueButton26 ( ) const [inline]
```

Get Blue Button 26.

References RDInput.

**9.126.6.24 GetBlueButton27()**

```
bool RaildriverIO::GetBlueButton27 ( ) const [inline]
```

Get Blue Button 27.

References RDInput.

**9.126.6.25 GetBlueButton28()**

```
bool RaildriverIO::GetBlueButton28 ( ) const [inline]
```

Get Blue Button 28.

References RDInput.

**9.126.6.26 GetBlueButton3()**

```
bool RaildriverIO::GetBlueButton3 ( ) const [inline]
```

Get Blue Button 3.

References RDInput.

**9.126.6.27 GetBlueButton4()**

```
bool RaildriverIO::GetBlueButton4 ( ) const [inline]
```

Get Blue Button 4.

References RDInput.

**9.126.6.28 GetBlueButton5()**

```
bool RaildriverIO::GetBlueButton5 ( ) const [inline]
```

Get Blue Button 5.

References RDInput.

**9.126.6.29 GetBlueButton6()**

```
bool RaildriverIO::GetBlueButton6 ( ) const [inline]
```

Get Blue Button 6.

References RDInput.

**9.126.6.30 GetBlueButton7()**

```
bool RaildriverIO::GetBlueButton7 ( ) const [inline]
```

Get Blue Button 7.

References RDInput.

**9.126.6.31 GetBlueButton8()**

```
bool RaildriverIO::GetBlueButton8 ( ) const [inline]
```

Get Blue Button 8.

References RDInput.

**9.126.6.32 GetBlueButton9()**

```
bool RaildriverIO::GetBlueButton9 ( ) const [inline]
```

Get Blue Button 9.

References RDInput.

**9.126.6.33 GetEBrakeDown()**

```
bool RaildriverIO::GetEBrakeDown ( ) const [inline]
```

Get Emergency Brake Down.

References RDInput.



**9.126.6.34 GetEBrakeUp()**

```
bool RaildriverIO::GetEBrakeUp ( ) const [inline]
```

Get Emergency Brake Up.

References RDInput.

**9.126.6.35 GetHeadlight()**

```
unsigned char RaildriverIO::GetHeadlight ( ) const [inline]
```

Get Headlight value (0-255).

References RDInput.

**9.126.6.36 GetIndependBrake()**

```
unsigned char RaildriverIO::GetIndependBrake ( ) const [inline]
```

Get Independant Brake value (0-255).

References RDInput.

**9.126.6.37 GetPanDown()**

```
bool RaildriverIO::GetPanDown ( ) const [inline]
```

Get Pan Down.

References RDInput.

**9.126.6.38 GetPanLeft()**

```
bool RaildriverIO::GetPanLeft ( ) const [inline]
```

Get Pan Left.

References RDInput.

**9.126.6.39 GetPanRight()**

```
bool RaildriverIO::GetPanRight ( ) const [inline]
```

Get Pan Right.

References RDInput.

**9.126.6.40 GetPantograph()**

```
bool RaildriverIO::GetPantograph ( ) const [inline]
```

Get Pantograph.

References RDInput.

**9.126.6.41 GetPanUp()**

```
bool RaildriverIO::GetPanUp ( ) const [inline]
```

Get Pan Up.

References RDInput.

**9.126.6.42 GetProductCodeId()**

```
unsigned char RaildriverIO::GetProductCodeId ( ) const [inline]
```

Get Product Code Id.

This is a unsigned char value filled in upon reading the input report buffer from the Raildriver.

References RDInput, and ReadInputs().

**9.126.6.43 GetRangeDown()**

```
bool RaildriverIO::GetRangeDown ( ) const [inline]
```

Get Range Down.

References RDInput.

**9.126.6.44 GetRangeUp()**

```
bool RaildriverIO::GetRangeUp ( ) const [inline]
```

Get Range Up.

References RDInput.

**9.126.6.45 GetReverser()**

```
unsigned char RaildriverIO::GetReverser ( ) const [inline]
```

Get Reverser value (0-255).

References RDInput.

**9.126.6.46 GetSand()**

```
bool RaildriverIO::GetSand ( ) const [inline]
```

Get Sand.

References RDInput.

**9.126.6.47 GetThrottle()**

```
unsigned char RaildriverIO::GetThrottle ( ) const [inline]
```

Get Throttle value (0-255).

References RDInput.

**9.126.6.48 GetWhistleDown()**

```
bool RaildriverIO::GetWhistleDown ( ) const [inline]
```

Get Whistle Down.

References RDInput.

**9.126.6.49 GetWhistleUp()**

```
bool RaildriverIO::GetWhistleUp ( ) const [inline]
```

Get Whistle Up.

References RDInput.

**9.126.6.50 GetWiper()**

```
unsigned char RaildriverIO::GetWiper ( ) const [inline]
```

Get Wiper value (0-255).

References RDInput.

**9.126.6.51 GetZoomUp()**

```
bool RaildriverIO::GetZoomUp ( ) const [inline]
```

Get Zoom Up.

References RDInput.

**9.126.6.52 GetZoomDown()**

```
bool RaildriverIO::GetZoomDown ( ) const [inline]
```

Get Zoom Down.

References RDInput.

**9.126.6.53 ReadInputs()**

```
bool RaildriverIO::ReadInputs (
    Eventmask_bits & mask,
    int & status )
```

Poll the interface.

Called in the event loop. Returns true if something has changed, that is if any of the bytes in the freshly read report buffer are different from the stored report buffer.

## Parameters

<i>mask</i>	Mask of changed bits. This parameter is updated to reflect any changed state information.
<i>status</i>	IO Status of the read.

Referenced by GetProductCodeId().

#### 9.126.6.54 SetLEDS()

```
void RaildriverIO::SetLEDS (
    const char * ledstring,
    char ** outmessage = NULL )
```

Set the Speedometer LEDs.

Does a bulk write to set the speedometer LEDs on the Raid Driver unit.

#### 9.126.6.55 SpeakerOff()

```
void RaildriverIO::SpeakerOff (
    char ** outmessage = NULL )
```

Turn the speaker off.

#### 9.126.6.56 SpeakerOn()

```
void RaildriverIO::SpeakerOn (
    char ** outmessage = NULL )
```

Turn the speaker on.

### 9.126.7 Member Data Documentation

#### 9.126.7.1 AutoBrake

```
unsigned char RaildriverIO::AutoBrake
```

Automatic Brake lever, 0-255.

**9.126.7.2 BailOff**

```
unsigned char RaildriverIO::BailOff
```

Bail Off (Independent Brake lever), 0-255.

**9.126.7.3 Digital1**

```
unsigned char RaildriverIO::Digital1
```

Blue Buttons 1-8.

This is an 8-bit bit field.

**9.126.7.4 Digital2**

```
unsigned char RaildriverIO::Digital2
```

Blue Buttons 9-16.

This is an 8-bit bit field.

**9.126.7.5 Digital3**

```
unsigned char RaildriverIO::Digital3
```

Blue Buttons 17-24.

This is an 8-bit bit field.

**9.126.7.6 Digital4**

```
unsigned char RaildriverIO::Digital4
```

Blue Buttons 25-28, Zoom, Pan Buttons.

This is an 8-bit bit field.

**9.126.7.7 Digital5**

```
unsigned char RaildriverIO::Digital5
```

Pan, Cab buttons.

This is an 8-bit bit field.

#### 9.126.7.8 Digital6

```
unsigned char RaildriverIO::Digital6
```

Cab Buttons, Whistle Switch.

This is an 8-bit bit field.

#### 9.126.7.9 Headlight

```
unsigned char RaildriverIO::Headlight
```

Headlight switch, 0-255.

#### 9.126.7.10 IndependBrake

```
unsigned char RaildriverIO::IndependBrake
```

Independent Brake lever, 0-255.

#### 9.126.7.11 LEDCommand

```
const int RaildriverIO::LEDCommand [static], [private]
```

LED Command code.

This is the command code used to change the LED display.

#### 9.126.7.12 PIEngineering

```
const unsigned short int RaildriverIO::PIEngineering [static], [private]
```

Rail Driver vendor code.

This is the vendor code PI Engineering was granted for their USB products.

Referenced by RaildriverIO().

**9.126.7.13 ProductCodeId**

```
unsigned char RaildriverIO::ProductCodeId
```

Product Code Id, usually 210.

**9.126.7.14 RailDriverModernDesktop**

```
const unsigned short int RaildriverIO::RailDriverModernDesktop [static], [private]
```

Rail Driver product code.

This is the product PI Engineering uses for their Rail Driver consoles.

**9.126.7.15 RDInput**

```
union { ... } RaildriverIO::RDInput [private]
```

Event data.

This is the report buffer used to hold the state of all of the levers, switches, and buttons on the Rail Driver console. It is a union of a fourteen byte buffer and a struct of bytes, one for each lever or switch or bit fields representing single buttons.

Referenced by GetAlert(), GetAutoBrake(), GetBailOff(), GetBell(), GetBlueButton1(), GetBlueButton10(), GetBlueButton11(), GetBlueButton12(), GetBlueButton13(), GetBlueButton14(), GetBlueButton15(), GetBlueButton16(), GetBlueButton17(), GetBlueButton18(), GetBlueButton19(), GetBlueButton2(), GetBlueButton20(), GetBlueButton21(), GetBlueButton22(), GetBlueButton23(), GetBlueButton24(), GetBlueButton25(), GetBlueButton26(), GetBlueButton27(), GetBlueButton28(), GetBlueButton3(), GetBlueButton4(), GetBlueButton5(), GetBlueButton6(), GetBlueButton7(), GetBlueButton8(), GetBlueButton9(), GetEBrakeDown(), GetEBrakeUp(), GetHeadlight(), GetIndependBrake(), GetPanDown(), GetPanLeft(), GetPanRight(), GetPantograph(), GetPanUp(), GetProductCodeId(), GetRangeDown(), GetRangeUp(), GetReverser(), GetSand(), GetThrottle(), GetWhistleDown(), GetWhistleUp(), GetWiper(), GetZoomUp(), and GetZoopDown().

**9.126.7.16 rdriverdev**

```
hid_device* RaildriverIO::rdriverdev [private]
```

Rail Driver Device.

This is the HID device structure for the device.



#### 9.126.7.17 ReportBuffer

```
unsigned char RaildriverIO::ReportBuffer[14]
```

Event Buffer.

This is the I/O buffer used to hold the information about the state of the Rail Driver console.

#### 9.126.7.18 Reverser

```
unsigned char RaildriverIO::Reverser
```

Reverser lever, 0-255.

#### 9.126.7.19 SpeakerCommand

```
const int RaildriverIO::SpeakerCommand [static], [private]
```

Speaker command code.

This is the command code used to toggle the speaker state.

#### 9.126.7.20 theBytes

```
struct { ... } ::bytes RaildriverIO::theBytes
```

#### 9.126.7.21 Throttle

```
unsigned char RaildriverIO::Throttle
```

Throttle / Dynamic Brake lever, 0-255.

#### 9.126.7.22 Wiper

```
unsigned char RaildriverIO::Wiper
```

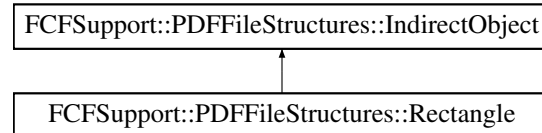
Wiper switch, 0-255.

## 9.127 FCFSupport::PDFFileStructures::Rectangle Class Reference

A rectangle object.

```
#include <PDFPrinterSupport.h>
```

Inheritance diagram for FCFSupport::PDFFileStructures::Rectangle:



### Public Member Functions

- [Rectangle](#) (double x\_1, double y\_1, double x\_2, double y\_2, unsigned long int objNum=0L, unsigned short int genNum=0, CrossReferenceTable \*tab=NULL)

*Constructor.*

- [~Rectangle](#) ()

*Destructor.*

- double [X1](#) () const

*Return the first X coordinate.*

- double [Y1](#) () const

*Return the first Y coordinate.*

- double [X2](#) () const

*Return the second X coordinate.*

- double [Y2](#) () const

*Return the second Y coordinate.*

- virtual ostream & [WriteDirect](#) (ostream &stream) const

*Write an object directly.*

### Private Attributes

- double [x1](#)

*First X coordinate.*

- double [y1](#)

*First Y coordinate.*

- double [x2](#)

*Second X coordinate.*

- double [y2](#)

*Second Y coordinate.*

### 9.127.1 Detailed Description

A rectangle object.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.127.2 Constructor & Destructor Documentation

#### 9.127.2.1 Rectangle()

```
FCFSupport::PDFFileStructures::Rectangle::Rectangle (
    double x_1,
    double y_1,
    double x_2,
    double y_2,
    unsigned long int objNum = 0L,
    unsigned short int genNum = 0,
    CrossReferenceTable * tab = NULL ) [inline]
```

Constructor.

Create a fresh [Rectangle](#) object.

#### Parameters

<i>x_1</i>	First X coordinate.
<i>y_1</i>	First Y coordinate.
<i>x_2</i>	Second X coordinate.
<i>y_2</i>	Second Y coordinate.
<i>objNum</i>	The next object number.
<i>genNum</i>	The generation number.
<i>tab</i>	The cross reference table we are in.

#### 9.127.2.2 ~Rectangle()

```
FCFSupport::PDFFileStructures::Rectangle::~~Rectangle ( ) [inline]
```

Destructor.

### 9.127.3 Member Function Documentation

#### 9.127.3.1 WriteDirect()

```
virtual ostream& FCFSupport::PDFFileStructures::Rectangle::WriteDirect (
    ostream & stream ) const [inline], [virtual]
```

Write an object directly.

##### Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

Implements [FCFSupport::PDFFileStructures::IndirectObject](#).

References `lcc::stream`.

#### 9.127.3.2 X1()

```
double FCFSupport::PDFFileStructures::Rectangle::X1 ( ) const [inline]
```

Return the first X coordinate.

#### 9.127.3.3 X2()

```
double FCFSupport::PDFFileStructures::Rectangle::X2 ( ) const [inline]
```

Return the second X coordinate.

#### 9.127.3.4 Y1()

```
double FCFSupport::PDFFileStructures::Rectangle::Y1 ( ) const [inline]
```

Return the first Y coordinate.

#### 9.127.3.5 Y2()

```
double FCFSupport::PDFFileStructures::Rectangle::Y2 ( ) const [inline]
```

Return the second Y coordinate.

### 9.127.4 Member Data Documentation

#### 9.127.4.1 x1

```
double FCFSupport::PDFFileStructures::Rectangle::x1 [private]
```

First X coordinate.

#### 9.127.4.2 x2

```
double FCFSupport::PDFFileStructures::Rectangle::x2 [private]
```

Second X coordinate.

#### 9.127.4.3 y1

```
double FCFSupport::PDFFileStructures::Rectangle::y1 [private]
```

First Y coordinate.

#### 9.127.4.4 y2

```
double FCFSupport::PDFFileStructures::Rectangle::y2 [private]
```

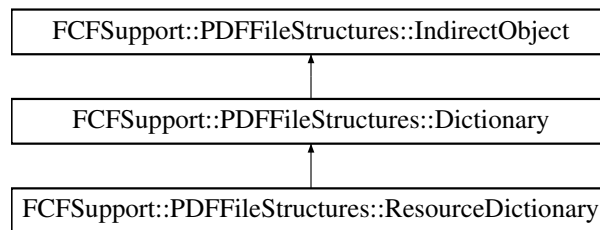
Second Y coordinate.

## 9.128 FCFSupport::PDFFileStructures::ResourceDictionary Class Reference

Resource dictionary.

```
#include <PDFPrinterSupport.h>
```

Inheritance diagram for FCFSupport::PDFFileStructures::ResourceDictionary:



### Public Member Functions

- [ResourceDictionary](#) (unsigned long int objNum=0L, unsigned short int genNum=0, CrossReferenceTable \*tab=↔ NULL)  
*Constructor.*
- [~ResourceDictionary](#) ()  
*Destructor.*
- void [AddProcSet](#) (string pname)  
*Add a procedure set.*
- void [AddExternalGraphicsState](#) (const string name, [IndirectObject](#) \*obj)  
*Add an External Graphics State resource.*
- void [AddColorSpace](#) (const string name, [IndirectObject](#) \*obj)  
*Add an Color Space resource.*
- void [AddPattern](#) (const string name, [IndirectObject](#) \*obj)  
*Add an Pattern resource.*
- void [AddShading](#) (const string name, [IndirectObject](#) \*obj)  
*Add an Shading resource.*
- void [AddXObject](#) (const string name, [IndirectObject](#) \*obj)  
*Add an External Object resource.*
- void [AddFont](#) (const string name, [IndirectObject](#) \*obj)  
*Add an Font resource.*
- void [AddProperties](#) (const string name, [IndirectObject](#) \*obj)  
*Add an Properties resource.*

### Protected Member Functions

- virtual ostream & [WriteDictionaryElements](#) (ostream &stream) const  
*Write the elements of a dictionary.*

## Private Attributes

- [IndirectObjectDictionary extGState](#)  
A dictionary that maps resource names to graphics state parameters dictionaries.
- [IndirectObjectDictionary colorSpace](#)  
A dictionary that maps each resource name to either the name of a device-dependent color space or to an array describing a color space.
- [IndirectObjectDictionary pattern](#)  
A *Dictionary* that maps resource names to pattern objects.
- [IndirectObjectDictionary shading](#)  
A *Dictionary* that maps resource names to shading dictionaries.
- [IndirectObjectDictionary xObject](#)  
A *Dictionary* that maps resource names to external objects.
- [IndirectObjectDictionary font](#)  
A *Dictionary* that maps resource names to font dictionaries.
- [IndirectObjectDictionary properties](#)  
A *Dictionary* that maps resource names to property list dictionaries for marked content.
- [PDFNameArray procSets](#)  
An array of predefined preseture set names.

### 9.128.1 Detailed Description

Resource dictionary.

Holds various named resources for indirect access.

Author

Robert Heller <heller@deepsoft.com>

### 9.128.2 Constructor & Destructor Documentation

#### 9.128.2.1 ResourceDictionary()

```
FCFSupport::PDFFileStructures::ResourceDictionary::ResourceDictionary (
    unsigned long int objNum = 0L,
    unsigned short int genNum = 0,
    CrossReferenceTable * tab = NULL ) [inline]
```

Constructor.

Create a new dictionary.

**Parameters**

<i>objNum</i>	The next object number.
<i>genNum</i>	The generation number.
<i>tab</i>	The cross reference table we are in.

**9.128.2.2    ~ResourceDictionary()**

```
FCFSupport::PDFFileStructures::ResourceDictionary::~~ResourceDictionary ( ) [inline]
```

Destructor.

**9.128.3    Member Function Documentation****9.128.3.1    AddColorSpace()**

```
void FCFSupport::PDFFileStructures::ResourceDictionary::AddColorSpace (
    const string name,
    IndirectObject * obj ) [inline]
```

Add an Color Space resource.

**Parameters**

<i>name</i>	The resource name.
<i>obj</i>	The indirect object.

**9.128.3.2    AddExternalGraphicsState()**

```
void FCFSupport::PDFFileStructures::ResourceDictionary::AddExternalGraphicsState (
    const string name,
    IndirectObject * obj ) [inline]
```

Add an External Graphics State resource.



## Parameters

<i>name</i>	The resource name.
<i>obj</i>	The indirect object.

## 9.128.3.3 AddFont()

```
void FCFSupport::PDFFileStructures::ResourceDictionary::AddFont (
    const string name,
    IndirectObject * obj ) [inline]
```

Add an Font resource.

## Parameters

<i>name</i>	The resource name.
<i>obj</i>	The indirect object.

## 9.128.3.4 AddPattern()

```
void FCFSupport::PDFFileStructures::ResourceDictionary::AddPattern (
    const string name,
    IndirectObject * obj ) [inline]
```

Add an Pattern resource.

## Parameters

<i>name</i>	The resource name.
<i>obj</i>	The indirect object.

## 9.128.3.5 AddProcSet()

```
void FCFSupport::PDFFileStructures::ResourceDictionary::AddProcSet (
    string pname ) [inline]
```

Add a procedure set.

**Parameters**

<i>pname</i>	The name of the prodecure set to add.
--------------	---------------------------------------

**9.128.3.6 AddProperties()**

```
void FCFSupport::PDFFileStructures::ResourceDictionary::AddProperties (
    const string name,
    IndirectObject * obj ) [inline]
```

Add an Properties resource.

**Parameters**

<i>name</i>	The resource name.
<i>obj</i>	The indirect object.

References lcc::stream.

**9.128.3.7 AddShading()**

```
void FCFSupport::PDFFileStructures::ResourceDictionary::AddShading (
    const string name,
    IndirectObject * obj ) [inline]
```

Add an Shading resource.

**Parameters**

<i>name</i>	The resource name.
<i>obj</i>	The indirect object.

**9.128.3.8 AddXObject()**

```
void FCFSupport::PDFFileStructures::ResourceDictionary::AddXObject (
    const string name,
    IndirectObject * obj ) [inline]
```

Add an External Object resource.

## Parameters

<i>name</i>	The resource name.
<i>obj</i>	The indirect object.

## 9.128.3.9 WriteDictionaryElements()

```
virtual ostream& FCFSupport::PDFFileStructures::ResourceDictionary::WriteDictionaryElements (
    ostream & stream ) const [protected], [virtual]
```

Write the elements of a dictionary.

## Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

Reimplemented from [FCFSupport::PDFFileStructures::Dictionary](#).

## 9.128.4 Member Data Documentation

## 9.128.4.1 colorSpace

```
IndirectObjectDictionary FCFSupport::PDFFileStructures::ResourceDictionary::colorSpace [private]
```

A dictionary that maps each resource name to either the name of a device-dependent color space or to an array describing a color space.

## 9.128.4.2 extGState

```
IndirectObjectDictionary FCFSupport::PDFFileStructures::ResourceDictionary::extGState [private]
```

A dictionary that maps resource names to graphics state parameters dictionaries.

#### 9.128.4.3 font

`IndirectObjectDictionary FCFSupport::PDFFileStructures::ResourceDictionary::font [private]`

A [Dictionary](#) that maps resource names to font dictionaries.

#### 9.128.4.4 pattern

`IndirectObjectDictionary FCFSupport::PDFFileStructures::ResourceDictionary::pattern [private]`

A [Dictionary](#) that maps resource names to pattern objects.

#### 9.128.4.5 procSets

`PDFNameArray FCFSupport::PDFFileStructures::ResourceDictionary::procSets [private]`

An array of predefined preseture set names.

#### 9.128.4.6 properties

`IndirectObjectDictionary FCFSupport::PDFFileStructures::ResourceDictionary::properties [private]`

A [Dictionary](#) that maps resource names to property list dictionaries for marked content.

#### 9.128.4.7 shading

`IndirectObjectDictionary FCFSupport::PDFFileStructures::ResourceDictionary::shading [private]`

A [Dictionary](#) that maps resource names to shading dictionaries.

#### 9.128.4.8 xObject

`IndirectObjectDictionary FCFSupport::PDFFileStructures::ResourceDictionary::xObject [private]`

A [Dictionary](#) that maps resource names to external objects.

## 9.129 Parsers::RouteVec Struct Reference

Route structure.

```
#include <TrackGraph.h>
```

### Public Attributes

- char \* [positionName](#)  
*Name of route.*
- [IntegerList](#) \* [posList](#)  
*List of segments used by the route.*
- float [routeLength](#)  
*Length of the route.*

### 9.129.1 Detailed Description

Route structure.

Author

Robert Heller <heller@deepsoft.com>

### 9.129.2 Member Data Documentation

#### 9.129.2.1 positionName

```
char* Parsers::RouteVec::positionName
```

Name of route.

#### 9.129.2.2 posList

```
IntegerList* Parsers::RouteVec::posList
```

List of segments used by the route.

### 9.129.2.3 routeLength

```
float Parsers::RouteVec::routeLength
```

Length of the route.

## 9.130 Satellite Class Reference

[Satellite](#) class.

### Public Member Functions

- [Satellite](#) (name, hostname,...)  
*Construct [Satellite](#) connection.*
- [remoteeval](#) (...)  
*Method to run a remote command.*
- [~Satellite](#) ()  
*Clean up all data objects and free up all resources.*

### Private Attributes

- [socket](#)  
*The connection socket.*

### 9.130.1 Detailed Description

[Satellite](#) class.

This class implements the interface logic to connect to a [Satellite](#) daemon running on a slave computer.

#### Parameters

<i>hostname</i>	The name or address of the slave.
...	Options: <ul style="list-style-type: none"><li>• -port The port to use to connect. Readonly, default is 40000.</li></ul>

#### Author

Robert Heller <heller@deepsoft.com>

## 9.130.2 Constructor & Destructor Documentation

### 9.130.2.1 Satellite()

```
Satellite::Satellite (
    name ,
    hostname ,
    ... )
```

Construct [Satellite](#) connection.

#### Parameters

<i>hostname</i>	The name or address of the slave.
...	Options: <ul style="list-style-type: none"><li>• -port The port to use to connect. Readonly, default is 40000.</li></ul>

### 9.130.2.2 ~Satellite()

```
Satellite::~~Satellite ( )
```

Clean up all data objects and free up all resources.

## 9.130.3 Member Function Documentation

### 9.130.3.1 remoteeval()

```
Satellite::remoteeval (
    ... )
```

Method to run a remote command.

**Parameters**

...	Command list.
-----	---------------

**Returns**

The result of the remote command.

**9.130.4 Member Data Documentation****9.130.4.1 socket**

`Satellite::socket` [private]

The connection socket.

**9.131 CTCPPanel::SchLabel Class Reference**

Schematic Label object type.

**Public Member Functions**

- [SchLabel](#) (name, \_ctcpanel, \_canvas,...)  
*Construct a Label object.*
- [~SchLabel](#) ()  
*Clean up all data objects and free up all resources.*
- [getv](#) ()  
*Method to get our value (none).*
- [setv](#) (state)  
*Method to set out value (level position).*
- [geti](#) (ind)  
*Method to get the state of one of our indicators (none).*
- [seti](#) (ind, value)  
*Method to set an indicator's state (none).*
- [invoke](#) ()  
*Method to invoke the label.*



## Private Member Functions

- [\\_configureColor](#) (option, value)  
*Method to update the color of the label.*
- [\\_configureLabel](#) (option, value)  
*Method to update the label option.*

## Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*

### 9.131.1 Detailed Description

Schematic Label object type.

These are on the schematic and represent a label on the Schematic.

#### Parameters

<a href="#">_ctcpanel</a>	The <a href="#">CTCPanel</a> megawidget.
<a href="#">_canvas</a>	The schematic canvas to draw the label on.
...	Options: <ul style="list-style-type: none"><li>• -x The x coordinate of the object (readonly, default 0).</li><li>• -y The y coordinate of the object (readonly, default 0).</li><li>• -controlpoint The name of the control point this label is part of (readonly, default CP1).</li><li>• -color The color of the label (default white).</li><li>• -label The label of the label (default "").</li></ul>

Defined coords terminals: none. Defined values (states): none. Defined indicators: none.

#### Author

Robert Heller <heller@deepsoft.com>

## 9.131.2 Constructor & Destructor Documentation

### 9.131.2.1 SchLabel()

```
CTCPanel::SchLabel::SchLabel (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a Label object.

#### Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the <a href="#">SchLabel</a> on.
<code>...</code>	Option list.

### 9.131.2.2 ~SchLabel()

```
CTCPanel::SchLabel::~~SchLabel ( )
```

Clean up all data objects and free up all resources.

## 9.131.3 Member Function Documentation

### 9.131.3.1 \_configureColor()

```
CTCPanel::SchLabel::_configureColor (
    option ,
    value ) [private]
```

Method to update the color of the label.

#### 9.131.3.2 \_configureLabel()

```
CTCPanel::SchLabel::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

#### 9.131.3.3 geti()

```
CTCPanel::SchLabel::geti (
    ind )
```

Method to get the state of one of our indicators (none).

#### 9.131.3.4 getv()

```
CTCPanel::SchLabel::getv ( )
```

Method to get our value (none).

#### 9.131.3.5 invoke()

```
CTCPanel::SchLabel::invoke ( )
```

Method to invoke the label.

#### 9.131.3.6 seti()

```
CTCPanel::SchLabel::seti (
    ind ,
    value )
```

Method to set an indicator's state (none).

### 9.131.3.7 setv()

```
CTCPanel::SchLabel::setv (
    state )
```

Method to set out value (level position).

## 9.131.4 Member Data Documentation

### 9.131.4.1 canvas

```
CTCPanel::SchLabel::canvas [private]
```

The canvas component (parent widget component).

### 9.131.4.2 ctcpnl

```
CTCPanel::SchLabel::ctcpnl [private]
```

The CTC Panel component (parent widget).

## 9.132 CTCPanel::ScissorCrossover Class Reference

Scissor [Crossover](#) (turnout) object type.

### Public Member Functions

- [ScissorCrossover](#) (name, \_ctcpnl, \_canvas,...)  
*Construct a [ScissorCrossover](#) object.*
- [~ScissorCrossover](#) ()  
*Clean up all data objects and free up all resources.*
- [getv](#) ()  
*Method to get our value (state).*
- [setv](#) (value)  
*Method to set out value (state).*
- [geti](#) (ind)  
*Method to get the state of one of our indicators (none).*
- [seti](#) (ind, value)  
*Method to set an indicator's state (none).*
- [invoke](#) ()  
*Method to invoke the switch.*

## Private Member Functions

- [\\_configureLabel](#) (option, value)  
*Method to update the label option.*

## Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*
- [state](#)  
*State of the points.*

### 9.132.1 Detailed Description

Scissor [Crossover](#) (turnout) object type.

These are on the schematic and represent a Scissor [Crossover](#) on the Schematic.

#### Parameters

<a href="#">_ctcpanel</a>	The <a href="#">CTCPanel</a> megawidget.
<a href="#">_canvas</a>	The schematic canvas to draw the switch on.
...	Options: <ul style="list-style-type: none"> <li>• -x The x coordinate of the object (readonly, default 0).</li> <li>• -y The y coordinate of the object (readonly, default 0).</li> <li>• -controlpoint The name of the control point this label is part of (readonly, default CP1).</li> <li>• -label The label of the switch (default "1").</li> <li>• -orientation The orientation (8-way) of the switch (readonly, default 0).</li> <li>• -flipped Whether or not the switch is flipped (readonly, default no).</li> <li>• -statecommand A command to run to get the switch's state (default {}).</li> <li>• -occupiedcommand A command to run to find out if the switch is occupied (default {}).</li> </ul>

Defined coords terminals:

- Main1L Upper left mainline.
- Main2L Lower left mainline.
- Main1R Upper right mainline.
- Main2R Lower right mainline.

Defined values (states):

- Normal Points are aligned for the mainline.
- Reverse Points are aligned for the branchline.
- Unknown Point are not aligned for any route (eg points are in motion).

Defined indicators: none.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.132.2 Constructor & Destructor Documentation

#### 9.132.2.1 ScissorCrossover()

```
CTCPanel::ScissorCrossover::ScissorCrossover (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a [ScissorCrossover](#) object.

#### Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the <a href="#">ScissorCrossover</a> on.
<code>...</code>	Option list.

#### 9.132.2.2 ~ScissorCrossover()

```
CTCPanel::ScissorCrossover::~~ScissorCrossover ( )
```

Clean up all data objects and free up all resources.

### 9.132.3 Member Function Documentation

#### 9.132.3.1 \_configureLabel()

```
CTCPanel::ScissorCrossover::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

#### 9.132.3.2 geti()

```
CTCPanel::ScissorCrossover::geti (
    ind )
```

Method to get the state of one of our indicators (none).

#### 9.132.3.3 getv()

```
CTCPanel::ScissorCrossover::getv ( )
```

Method to get our value (state).

#### 9.132.3.4 invoke()

```
CTCPanel::ScissorCrossover::invoke ( )
```

Method to invoke the switch.

#### 9.132.3.5 seti()

```
CTCPanel::ScissorCrossover::seti (
    ind ,
    value )
```

Method to set an indicator's state (none).

#### 9.132.3.6 setv()

```
CTCPanel::ScissorCrossover::setv (
    value )
```

Method to set out value (state).

**Parameters**

<i>value</i>	The new state to set.
--------------	-----------------------

**9.132.4 Member Data Documentation****9.132.4.1 canvas**

```
CTCPanel::ScissorCrossover::canvas [private]
```

The canvas component (parent widget component).

**9.132.4.2 ctcpnl**

```
CTCPanel::ScissorCrossover::ctcpnl [private]
```

The CTC Panel component (parent widget).

**9.132.4.3 state**

```
CTCPanel::ScissorCrossover::state [private]
```

State of the points.

**9.133 ScrollTabNotebook Class Reference**

Tabbed Notebook with scrollable tabs.



## Public Member Functions

- [ScrollTabNotebook](#) (name,...)  
*Constructor: create a [ScrollTabNotebook](#).*
- [compute\\_size](#) ()  
*(Re-)compute the size of the widget.*
- [add](#) (window,...)  
*Add a window to the end of the page list.*
- [insert](#) (pos, window,...)  
*Insert a window at the specified position.*
- [tab](#) (tabid,...)  
*Updates the tab options for tabid.*
- [tabs](#) ()  
*Return all tabs.*
- [forget](#) (tabid)  
*Removes the tab specified by tabid, unmaps and unmanages the associated window.*
- [select](#) (tabid="")  
*Selects the specified tab.*
- [see](#) (tabid)  
*Make the specified tabid visible.*
- [index](#) (tabid)  
*Returns the numeric index of the tab specified by tabid, or the total number of tabs if tabid is the string "end".*

## Private Member Functions

- [\\_test\\_page](#) (page)  
*Test to see if page is managed.*
- [\\_compute\\_width](#) ()  
*Recompute tab width.*
- [\\_compute\\_height](#) ()  
*Recompute tab height.*
- [\\_get\\_x\\_page](#) (pos)  
*Get X position of the page at pos.*
- [\\_xview](#) (inc)  
*Shift the tabs to the left or right.*
- [\\_highlight](#) (flag, page)  
*Turn highlighting on or off for the specified tab.*
- [\\_select](#) (page)  
*Make the specified page the selected page.*
- [\\_redraw](#) ()  
*Redraw the tabs and all.*
- [\\_draw\\_page](#) (page, create)  
*Draw a tab for the specified page, creating a new tab if asked or moving an old one otherwise.*
- [\\_draw\\_arrows](#) ()  
*Draw the arrow buttons if needed.*
- [\\_resize](#) ()  
*Resize the widget.*
- [\\_themeChanged](#) ()  
*Theme Changed method.*

## Static Private Member Functions

- static [\\_themeChanged](#) (w)  
*Theme Changed typemethod.*
- static [\\_Configure](#) (widget, width, height)  
*Configure typemethod.*
- static [get3dcolor](#) (path, bgcolor)  
*Compute 3D colors.*

## Private Attributes

- [tabrow](#)  
*Row containing the tabs.*
- [left](#)  
*Left arrow button.*
- [tabs](#)  
*Scrolling tab frame (canvas).*
- [right](#)  
*Right arrow button.*
- [select](#)  
*The currently selected page.*
- [\\_clientRow](#)  
*Grid row for client pages (1 is bottom).*
- [\\_tabrow](#)  
*Grid row for tabs (0 is top).*
- [\\_hpage](#)  
*Height of tabrow.*
- [\\_wpage](#)  
*Width of tabrow.*
- [\\_textid](#)  
*Scratch text id.*
- [realized](#)  
*Flag to indicate if the widget is realized.*
- [pages](#)  
*The list of available pages.*
- [pages\\_opts](#)  
*The options for the available pages.*
- [base](#)  
*Leftmost visible tab.*
- [dbg](#)  
*Darker tab background.*
- [lbg](#)  
*Lighter tab background.*

## Static Private Attributes

- static [\\_tabsides](#)  
*Tabside type checker.*
- static [\\_radiustype](#)  
*Tab radius and bevelsize type checker.*
- static [\\_paddingtype](#)  
*Tab padding type checker.*
- static [\\_warrow](#)  
*Width of an arrow button.*
- static [\\_left](#)  
*Bitmap for the left button.*
- static [\\_right](#)  
*Bitmap for the right button.*

### 9.133.1 Detailed Description

Tabbed Notebook with scrollable tabs.

This is a Tabbed Notebook widget, with scrollable tabs. It implements left and right arrows, as needed, to shift the tab row to the left or right to allow for more tabs than will fit in the available space.

Options:

- `-style` The style to use. The default is [ScrollTabNotebook](#).
- `-width` The width in pixels.
- `-height` The height in pixels.
- `-cursor` The cursor to use.
- `-takefocus` Can the widget take focus?

### 9.133.2 Constructor & Destructor Documentation

#### 9.133.2.1 ScrollTabNotebook()

```
ScrollTabNotebook::ScrollTabNotebook (  
    name ,  
    ... )
```

Constructor: create a [ScrollTabNotebook](#).

## Parameters

<i>name</i>	Pathname of the widget.
...	<div>Options:</div> <ul style="list-style-type: none"><li>• -style Widget style.</li><li>• -width The width of the widget.</li><li>• -height The height of the widget.</li><li>• -cursor The cursor to use.</li><li>• -takefocus Can the widget take focus?</li></ul>

### 9.133.3 Member Function Documentation

#### 9.133.3.1 \_compute\_height()

```
ScrollTabNotebook::_compute_height ( ) [private]
```

Recompute tab height.

#### 9.133.3.2 \_compute\_width()

```
ScrollTabNotebook::_compute_width ( ) [private]
```

Recompute tab width.

#### 9.133.3.3 \_Configure()

```
static ScrollTabNotebook::_Configure (
    widget ,
    width ,
    height ) [static], [private]
```

Configure typemethod.

## Parameters

<i>widget</i>	The widget the Configure event happened for.
<i>width</i>	The new width.
<i>height</i>	The new height.

9.133.3.4 `_draw_arrows()`

```
ScrollTabNotebook::_draw_arrows ( ) [private]
```

Draw the arrow buttons if needed.

9.133.3.5 `_draw_page()`

```
ScrollTabNotebook::_draw_page (
    page ,
    create ) [private]
```

Draw a tab for the specified page, creating a new tab if asked or moving an old one otherwise.

## Parameters

<i>page</i>	The page whose tab we will draw.
<i>create</i>	Flag indicating if the tab needs to be created.

9.133.3.6 `_get_x_page()`

```
ScrollTabNotebook::_get_x_page (
    pos ) [private]
```

Get X position of the page at pos.

## Parameters

<i>pos</i>	The page position.
------------	--------------------

**Returns**

The x position of the tab.

**9.133.3.7 \_highlight()**

```
ScrollTabNotebook::_highlight (
    flag ,
    page ) [private]
```

Turn highlighting on or off for the specified tab.

**Parameters**

<i>flag</i>	Flag to indicate turning highlighting on or off.
<i>page</i>	The page whose tab to turn highlighting on or off.

**9.133.3.8 \_redraw()**

```
ScrollTabNotebook::_redraw ( ) [private]
```

Redraw the tabs and all.

**9.133.3.9 \_resize()**

```
ScrollTabNotebook::_resize ( ) [private]
```

Resize the widget.

**9.133.3.10 \_select()**

```
ScrollTabNotebook::_select (
    page ) [private]
```

Make the specified page the selected page.

## Parameters

<i>page</i>	The page to select.
-------------	---------------------

9.133.3.11 `_test_page()`

```
ScrollTabNotebook::_test_page (  
    page ) [private]
```

Test to see if page is managed.

## Parameters

<i>page</i>	The page to test.
-------------	-------------------

## Returns

The position of the page.

9.133.3.12 `_themeChanged()`

```
static ScrollTabNotebook::_themeChanged (  
    w ) [static], [private]
```

Theme Changed typemethod.

## Parameters

<i>w</i>	The widget the theme changed for.
----------	-----------------------------------

9.133.3.13 `_themeChanged_()`

```
ScrollTabNotebook::_themeChanged_ ( ) [private]
```

Theme Changed method.

#### 9.133.3.14 `_xview()`

```
ScrollTabNotebook::_xview (  
    inc ) [private]
```

Shift the tabs to the left or right.

##### Parameters

<i>inc</i>	The shift increment, negative to the left, positive to the right.
------------	---

#### 9.133.3.15 `add()`

```
ScrollTabNotebook::add (  
    window ,  
    ... )
```

Add a window to the end of the page list.

Adds a new window (page) to the list of managed pages.

##### Parameters

<i>window</i>	The window to add.
...	<p>Tab options:</p> <ul style="list-style-type: none"><li>• <code>-state</code> The state of the tab (NOT IMPLEMENTED - state is always normal).</li><li>• <code>-sticky</code> The stickyness (as in grid configure ... <code>-sticky</code>).</li><li>• <code>-padding</code> The padding (as in grid configure ... <code>-padx</code> and <code>-pady</code>).</li><li>• <code>-text</code> The text of the tab.</li><li>• <code>-image</code> The image of the tab.</li><li>• <code>-compound</code> The compound of the tab (see the <code>-compound</code> option of labels and buttons).</li><li>• <code>-underline</code> The underline of the tab label (NOT IMPLEMENTED, the <code>-underline</code> option is ignored).</li></ul>



#### 9.133.3.16 compute\_size()

```
ScrollTabNotebook::compute_size ( )
```

(Re-)compute the size of the widget.

#### 9.133.3.17 forget()

```
ScrollTabNotebook::forget (
    tabid )
```

Removes the tab specified by tabid, unmaps and unmanages the associated window.

##### Parameters

<i>tabid</i>	The tab to remove.
--------------	--------------------

#### 9.133.3.18 get3dcolor()

```
static ScrollTabNotebook::get3dcolor (
    path ,
    bgcolor ) [static], [private]
```

Compute 3D colors.

##### Parameters

<i>path</i>	Window path.
<i>bgcolor</i>	Background color to use as a base.

##### Returns

Two RGB colors, one darker, one lighter.

#### 9.133.3.19 index()

```
ScrollTabNotebook::index (
    tabid )
```

Returns the numeric index of the tab specified by tabid, or the total number of tabs if tabid is the string "end".

## Parameters

<i>tabid</i>	The tabid to get the index of.
--------------	--------------------------------

## Returns

The numeric index of the tab specified by tabid.

## 9.133.3.20 insert()

```
ScrollTabNotebook::insert (
    pos ,
    window ,
    ... )
```

Insert a window at the specified position.

Inserts a new window (page) to the list of managed pages at the specified position.

## Parameters

<i>pos</i>	The insert position.
<i>window</i>	The window to insert.
...	Tab options: <ul style="list-style-type: none"> <li>• <i>-state</i> The state of the tab (NOT IMPLEMENTED - state is always normal).</li> <li>• <i>-sticky</i> The stickyness (as in grid configure ... <i>-sticky</i>).</li> <li>• <i>-padding</i> The padding (as in grid configure ... <i>-padx</i> and <i>-pady</i>).</li> <li>• <i>-text</i> The text of the tab.</li> <li>• <i>-image</i> The image of the tab.</li> <li>• <i>-compound</i> The compound of the tab (see the <i>-compound</i> option of labels and buttons).</li> <li>• <i>-underline</i> The underline of the tab label (NOT IMPLEMENTED, the <i>-underline</i> option is ignored).</li> </ul>

## 9.133.3.21 see()

```
ScrollTabNotebook::see (
    tabid )
```

Make the specified tabid visible.

#### Parameters

<i>tabid</i>	The tabid to make visible.
--------------	----------------------------

#### 9.133.3.22 select()

```
ScrollTabNotebook::select (
    tabid = "" )
```

Selects the specified tab.

The associated slave window will be displayed, and the previously-selected window (if different) is unmapped. If tabid is omitted, returns the widget name of the currently selected pane.

#### Parameters

<i>tabid</i>	The tab to select.
--------------	--------------------

#### Returns

If tabid is omitted, return the currently selected pane.

#### 9.133.3.23 tab()

```
ScrollTabNotebook::tab (
    tabid ,
    ... )
```

Updates the tab options for tabid.

#### Parameters

<i>tabid</i>	The tab index.
--------------	----------------

## Parameters

...	<p>Tab options:</p> <ul style="list-style-type: none"><li>• -state The state of the tab (NOT IMPLEMENTED - state is always normal).</li><li>• -sticky The stickyness (as in grid configure ... -sticky).</li><li>• -padding The padding (as in grid configure ... -padx and -pady).</li><li>• -text The text of the tab.</li><li>• -image The image of the tab.</li><li>• -compound The compound of the tab (see the -compound option of labels and buttons).</li><li>• -underline The underline of the tab label (NOT IMPLEMENTED, the -underline option is ignored).</li></ul>
-----	--

**9.133.3.24 tabs()**

```
ScrollTabNotebook::tabs ( )
```

Return all tabs.

**Returns**

All managed windows

**9.133.4 Member Data Documentation****9.133.4.1 \_clientRow**

```
ScrollTabNotebook::_clientRow [private]
```

Grid row for client pages (1 is bottom).

#### 9.133.4.2 \_hpage

`ScrollTabNotebook::_hpage` [private]

Height of tabrow.

#### 9.133.4.3 \_left

`ScrollTabNotebook::_left` [static], [private]

Bitmap for the left button.

#### 9.133.4.4 \_paddingtype

`ScrollTabNotebook::_paddingtype` [static], [private]

Tab padding type checker.

#### 9.133.4.5 \_radiustype

`ScrollTabNotebook::_radiustype` [static], [private]

Tab radius and bevelsize type checker.

#### 9.133.4.6 \_right

`ScrollTabNotebook::_right` [static], [private]

Bitmap for the right button.

#### 9.133.4.7 \_tabrow

`ScrollTabNotebook::_tabrow` [private]

Grid row for tabs (0 is top).

#### 9.133.4.8 \_tabsides

`ScrollTabNotebook::_tabsides [static], [private]`

Tabside type checker.

#### 9.133.4.9 \_textid

`ScrollTabNotebook::_textid [private]`

Scratch text id.

#### 9.133.4.10 \_warrow

`ScrollTabNotebook::_warrow [static], [private]`

Width of an arrow button.

#### 9.133.4.11 \_wpage

`ScrollTabNotebook::_wpage [private]`

Width of tabrow.

#### 9.133.4.12 base

`ScrollTabNotebook::base [private]`

Leftmost visible tab.

#### 9.133.4.13 dbg

`ScrollTabNotebook::dbg [private]`

Darker tab background.

**9.133.4.14 lbg**

```
ScrollTabNotebook::lbg [private]
```

Lighter tab background.

**9.133.4.15 left**

```
ScrollTabNotebook::left [private]
```

Left arrow button.

**9.133.4.16 pages**

```
ScrollTabNotebook::pages [private]
```

The list of available pages.

**9.133.4.17 pages\_opts**

```
ScrollTabNotebook::pages_opts [private]
```

The options for the available pages.

**9.133.4.18 realized**

```
ScrollTabNotebook::realized [private]
```

Flag to indicate if the widget is realized.

**9.133.4.19 right**

```
ScrollTabNotebook::right [private]
```

Right arrow button.

#### 9.133.4.20 select

```
ScrollTabNotebook::select [private]
```

The currently selected page.

#### 9.133.4.21 tabrow

```
ScrollTabNotebook::tabrow [private]
```

Row containing the tabs.

#### 9.133.4.22 tabs

```
ScrollTabNotebook::tabs [private]
```

Scrolling tab frame (canvas).

## 9.134 Parsers::SegPos Struct Reference

Segment position, endpoint or other coordinate.

```
#include <TrackGraph.h>
```

### Public Attributes

- float [x](#)  
*\$X\$ coordinate.*
- float [y](#)  
*\$Y\$ coordinate.*

### 9.134.1 Detailed Description

Segment position, endpoint or other coordinate.

#### Author

Robert Heller <heller@deepsoft.com>



### 9.134.2 Member Data Documentation

#### 9.134.2.1 x

```
float Parsers::SegPos::x
```

\$X\$ coordinate.

#### 9.134.2.2 y

```
float Parsers::SegPos::y
```

\$Y\$ coordinate.

## 9.135 Parsers::SegVector Struct Reference

Segemnt structure.

```
#include <TrackGraph.h>
```

### Public Types

- enum [GrType](#) { [S](#), [C](#), [J](#) }  
*Graphic types.*

### Public Attributes

- [GrType](#) [tgType](#)  
*Segment type.*
- [SegPos](#) [gPos1](#)  
*First graphic position.*
- [SegPos](#) [gPos2](#)  
*Second graphic position.*
- [SegPos](#) [ePos1](#)  
*First end point position.*
- [SegPos](#) [ePos2](#)  
*Second end point position.*
- float [radius](#)  
*Radius value.*

- float [ang0](#)  
*First angle.*
- float [ang1](#)  
*Second angle.*
- float [R](#)  
*\$R\$ value.*
- float [L](#)  
*\$L\$ value.*
- float [angle](#)  
*An angle.*
- float [len0](#)  
*First length parameter.*
- float [len1](#)  
*Second length parameter.*
- float [length](#)  
*Length of segment.*

### 9.135.1 Detailed Description

Segemnt structure.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.135.2 Member Enumeration Documentation

#### 9.135.2.1 GrType

enum [Parsers::SegVector::GrType](#)

Graphic types.

#### Enumerator

S	Straight segment.
C	Curved (circular) segment.
J	Curved (spiral easement) segment.

### 9.135.3 Member Data Documentation

#### 9.135.3.1 ang0

`float Parsers::SegVector::ang0`

First angle.

#### 9.135.3.2 ang1

`float Parsers::SegVector::ang1`

Second angle.

#### 9.135.3.3 angle

`float Parsers::SegVector::angle`

An angle.

#### 9.135.3.4 ePos1

`SegPos Parsers::SegVector::ePos1`

First end point position.

#### 9.135.3.5 ePos2

`SegPos Parsers::SegVector::ePos2`

Second end point position.

#### 9.135.3.6 gPos1

`SegPos` `Parsers::SegVector::gPos1`

First graphic position.

#### 9.135.3.7 gPos2

`SegPos` `Parsers::SegVector::gPos2`

Second graphic position.

#### 9.135.3.8 L

`float` `Parsers::SegVector::L`

\$L\$ value.

#### 9.135.3.9 len0

`float` `Parsers::SegVector::len0`

First length parameter.

#### 9.135.3.10 len1

`float` `Parsers::SegVector::len1`

Second length parameter.

#### 9.135.3.11 length

`float` `Parsers::SegVector::length`

Length of segment.

## 9.135.3.12 R

```
float Parsers::SegVector::R
```

\$R\$ value.

## 9.135.3.13 radius

```
float Parsers::SegVector::radius
```

Radius value.

## 9.135.3.14 tgType

```
GrType Parsers::SegVector::tgType
```

Segment type.

## 9.136 CabWidgets::SelectLocomotive Class Reference

Select or enter a Locomotive address.

### Public Member Functions

- [SelectLocomotive](#) (name,...)  
*Constructor.*
- [currentLocomotive](#) ()  
*Method to return the current locomotive address.*
- [invoke](#) ()  
*Method to invoke the widget.*

### Private Member Functions

- [\\_trimList](#) (option, value)  
*Configure method for -maxsaved.*
- [\\_addnewloco](#) ()  
*Add new loco.*

## Private Attributes

- [lf](#)  
*LabelFrame component.*
- [locoList](#)  
*Locolist Combobox component.*

### 9.136.1 Detailed Description

Select or enter a Locomotive address.

This widget implements a Locomotive address selection widget. A Locomotive is selected from a drop down or a new address is entered. When a new address is entered, it is saved in the drop down list. The maximum number of saved addresses is configurable.

#### Parameters

<i>path</i>	Pathname of the widget.
...	<p>Options:</p> <ul style="list-style-type: none"><li>• <code>-command</code> Script to call when the address is changed. The new address is appended.</li><li>• <code>-maxsaved</code> The maximum number of addresses to save. Default 6.</li><li>• <code>-label</code> The label to use.</li><li>• <code>-labelwidth</code> The width of the label.</li><li>• <code>-defaultlist</code> The list of default loco addresses. Readonly. Default {3}.</li></ul>

#### Author

Robert Heller <heller@deepsoft.com>

### 9.136.2 Constructor & Destructor Documentation

#### 9.136.2.1 SelectLocomotive()

```
CabWidgets::SelectLocomotive::SelectLocomotive (  
    name ,  
    ... )
```

Constructor.

## Parameters

<i>path</i>	Widget path.
...	Options.

### 9.136.3 Member Function Documentation

#### 9.136.3.1 \_addnewloco()

```
CabWidgets::SelectLocomotive::_addnewloco ( ) [private]
```

Add new loco.

Bound to the locoList ComboBox entry.

#### 9.136.3.2 \_trimList()

```
CabWidgets::SelectLocomotive::_trimList (
    option ,
    value ) [private]
```

Configure method for -maxsaved.

Trim the list if needed.

## Parameters

<i>option</i>	The option name.
<i>value</i>	The new value.

#### 9.136.3.3 currentLocomotive()

```
CabWidgets::SelectLocomotive::currentLocomotive ( )
```

Method to return the current locomotive address.

#### 9.136.3.4 invoke()

```
CabWidgets::SelectLocomotive::invoke ( )
```

Method to invoke the widget.

This calls the script (if any) defined by the -command option.

### 9.136.4 Member Data Documentation

#### 9.136.4.1 lf

```
CabWidgets::SelectLocomotive::lf [private]
```

LabelFrame component.

#### 9.136.4.2 locoList

```
CabWidgets::SelectLocomotive::locoList [private]
```

Locolist Combobox component.

## 9.137 Icc::SendEvent Class Reference

Send Event Dialog – send PCRE message.

### Public Member Functions

- [SendEvent](#) (name,...)  
*Construct a [SendEvent](#) dialog.*
- [\\_Close](#) ()  
*Close the window.*
- [\\_Send](#) ()  
*Bound to the *Send* button.*

### Private Attributes

- [eventid](#)  
*LabelEntry containing the eventId.*



### 9.137.1 Detailed Description

Send Event Dialog – send PCRE message.

Options:

- -transport The transport to use.

### 9.137.2 Constructor & Destructor Documentation

#### 9.137.2.1 SendEvent()

```
lcc::SendEvent::SendEvent (
    name ,
    ... )
```

Construct a [SendEvent](#) dialog.

##### Parameters

<i>name</i>	Pathname of the widget.
...	Options: <ul style="list-style-type: none"><li>• -transport LCC Transport object.</li></ul>

### 9.137.3 Member Function Documentation

#### 9.137.3.1 \_Close()

```
lcc::SendEvent::_Close ( )
```

Close the window.

### 9.137.3.2 `_Send()`

```
lcc::SendEvent::_Send ( )
```

Bound to the `Send` button.

Send an event.

## 9.137.4 Member Data Documentation

### 9.137.4.1 `eventid`

```
lcc::SendEvent::eventid [private]
```

LabelEntry containing the eventId.

## 9.138 `xpressnet::ServiceModeResponse` Class Reference

Service mode response.

### Public Member Functions

- [ServiceModeResponse](#) (name, modebits, CE, D)  
*Constructor.*
- [ServiceMode](#) ()  
*Return the service mode.*
- [CV](#) ()  
*Return the CV value.*
- [Data](#) ()  
*Return the data value.*

### Private Attributes

- [\\_service\\_mode](#)  
*The service mode.*
- [\\_cv](#)  
*The CV value.*
- [\\_data](#)  
*The data value.*

### 9.138.1 Detailed Description

Service mode response.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.138.2 Constructor & Destructor Documentation

#### 9.138.2.1 ServiceModeResponse()

```
xpressnet::ServiceModeResponse::ServiceModeResponse (
    name ,
    modebits ,
    CE ,
    D )
```

Constructor.

#### Parameters

<i>modebits</i>	First data byte (contains mode bit).
<i>CE</i>	Second data byte (contains C or E value).
<i>D</i>	Third data byte (contains D value).

### 9.138.3 Member Function Documentation

#### 9.138.3.1 CV()

```
xpressnet::ServiceModeResponse::CV ( )
```

Return the CV value.

#### 9.138.3.2 Data()

```
xpressnet::ServiceModeResponse::Data ( )
```

Return the data value.

### 9.138.3.3 ServiceMode()

```
xpressnet::ServiceModeResponse::ServiceMode ( )
```

Return the service mode.

## 9.138.4 Member Data Documentation

### 9.138.4.1 \_cv

```
xpressnet::ServiceModeResponse::_cv [private]
```

The CV value.

### 9.138.4.2 \_data

```
xpressnet::ServiceModeResponse::_data [private]
```

The data value.

### 9.138.4.3 \_service\_mode

```
xpressnet::ServiceModeResponse::_service_mode [private]
```

The service mode.

## 9.139 FCFSupport::ShowBannerCallback Class Reference

Display a page heading type message on the screen.

```
#include <CallBack.h>
```

## Public Member Functions

- [ShowBannerCallback](#) ()  
*Constructor.*
- virtual [~ShowBannerCallback](#) ()  
*Destructor.*
- virtual void [ShowBanner](#) () const  
*Display the application supplied banner text.*

### 9.139.1 Detailed Description

Display a page heading type message on the screen.

This callback simply has the application display its banner text identifying itself. Usually called before a series of informational messages relating to the progress of the processing.

@author Robert Heller \<heller\@deepsoft.com\>

### 9.139.2 Constructor & Destructor Documentation

#### 9.139.2.1 ShowBannerCallback()

```
FCFSupport::ShowBannerCallback::ShowBannerCallback ( ) [inline]
```

Constructor.

The base constructor does nothing. It is presumed that a derived class might do something useful.

#### 9.139.2.2 ~ShowBannerCallback()

```
virtual FCFSupport::ShowBannerCallback::~~ShowBannerCallback ( ) [inline], [virtual]
```

Destructor.

The base destructor does nothing. It is presumed that a derived class might do something useful.

### 9.139.3 Member Function Documentation

### 9.139.3.1 ShowBanner()

```
virtual void FCFSupport::ShowBannerCallback::ShowBanner ( ) const [inline], [virtual]
```

Display the application supplied banner text.

## 9.140 CTCPanel::Signal Class Reference

Signal object type.

### Public Member Functions

- [Signal](#) (name, \_ctcpanel, \_canvas,...)  
*Construct a [Signal](#) object.*
- [~Signal](#) ()  
*Clean up all data objects and free up all resources.*
- [getv](#) ()  
*Method to get our value (aspect).*
- [setv](#) (value)  
*Method to set out value (state).*
- [geti](#) (ind)  
*Method to get the state of one of our indicators (none).*
- [seti](#) (ind, value)  
*Method to set an indicator's state (none).*
- [invoke](#) ()  
*Method to invoke the [Signal](#).*

### Private Member Functions

- [\\_configureLabel](#) (option, value)  
*Method to update the label option.*
- [\\_VerifyHeads](#) (option, value)

### Static Private Member Functions

- static [\\_SchematicDrawThinLine](#) (canvas, x1, y1, x2, y2, orientation, tags)  
*Typemethod to draw a thin line.*
- static [\\_SchematicDrawOval](#) (canvas, x1, y1, dia, orientation, tags)  
*Typemethod to draw an invisible dot on the trackwork.*

## Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*
- [aspect](#)  
*Contains the signal's aspect.*

### 9.140.1 Detailed Description

Signal object type.

These are on the schematic and represent a [Signal](#) on the Schematic.

#### Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the <a href="#">Signal</a> on.
<code>...</code>	Options: <ul style="list-style-type: none"><li>• <code>-x</code> The x coordinate of the object (readonly, default 0).</li><li>• <code>-y</code> The y coordinate of the object (readonly, default 0).</li><li>• <code>-controlpoint</code> The name of the control point this label is part of (readonly, default CP1).</li><li>• <code>-label</code> The label of the <a href="#">Signal</a> (default "1").</li><li>• <code>-orientation</code> The orientation (8-way) of the <a href="#">Signal</a> (readonly, default 0).</li><li>• <code>-heads</code> The number of heads (1, 2, or 3) of the <a href="#">Signal</a> (readonly, default 1)</li></ul>

Defined coords terminals: none. Defined values: The signal's aspect. Defined indicators: none.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.140.2 Constructor & Destructor Documentation

### 9.140.2.1 Signal()

```
CTCPanel::Signal::Signal (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a [Signal](#) object.

#### Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the <a href="#">Signal</a> on.
<code>...</code>	Option list.

### 9.140.2.2 ~Signal()

```
CTCPanel::Signal::~~Signal ( )
```

Clean up all data objects and free up all resources.

## 9.140.3 Member Function Documentation

### 9.140.3.1 \_configureLabel()

```
CTCPanel::Signal::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

### 9.140.3.2 \_SchematicDrawOval()

```
static CTCPanel::Signal::_SchematicDrawOval (
    canvas ,
    x1 ,
    y1 ,
    dia ,
    orientation ,
    tags ) [static], [private]
```

Typemethod to draw an invisible dot on the trackwork.

Used as anchor points to connect trackwork sections together.



## Parameters

<i>x1</i>	The first X coordinate.
<i>y1</i>	The first Y coordinate.
<i>dia</i>	The diameter of the oval.
<i>orientation</i>	The orientation (8-way).
<i>tags</i>	The canvas tags to include.

## 9.140.3.3 \_SchematicDrawThinLine()

```
static CTCPanel::Signal::_SchematicDrawThinLine (  
    canvas ,  
    x1 ,  
    y1 ,  
    x2 ,  
    y2 ,  
    orientation ,  
    tags ) [static], [private]
```

Type method to draw a thin line.

## Parameters

<i>canvas</i>	The canvas to draw on.
<i>x1</i>	The first X coordinate.
<i>y1</i>	The first Y coordinate.
<i>x2</i>	The second X coordinate.
<i>y2</i>	The second Y coordinate.
<i>orientation</i>	The orientation (8-way).
<i>tags</i>	The canvas tags to include.

## 9.140.3.4 \_VerifyHeads()

```
CTCPanel::Signal::_VerifyHeads (  
    option ,  
    value ) [private]
```

**9.140.3.5    geti()**

```
CTCPanel::Signal::geti (
    ind )
```

Method to get the state of one of our indicators (none).

**9.140.3.6    getv()**

```
CTCPanel::Signal::getv ( )
```

Method to get our value (aspect).

**9.140.3.7    invoke()**

```
CTCPanel::Signal::invoke ( )
```

Method to invoke the [Signal](#).

**9.140.3.8    seti()**

```
CTCPanel::Signal::seti (
    ind ,
    value )
```

Method to set an indicator's state (none).

**9.140.3.9    setv()**

```
CTCPanel::Signal::setv (
    value )
```

Method to set out value (state).

**Parameters**

<i>value</i>	The new state to set.
--------------	-----------------------

### 9.140.4 Member Data Documentation

#### 9.140.4.1 aspect

CTCPanel::Signal::aspect [private]

Contains the signal's aspect.

#### 9.140.4.2 canvas

CTCPanel::Signal::canvas [private]

The canvas component (parent widget component).

#### 9.140.4.3 ctcpnl

CTCPanel::Signal::ctcpnl [private]

The CTC Panel component (parent widget).

## 9.141 CTCPanel::SIGPlate Class Reference

Signal plate object type.

### Public Member Functions

- [SIGPlate](#) (name, \_ctcpnl, \_canvas,...)  
*Construct a [SIGPlate](#) object.*
- [~SIGPlate](#) ()  
*Clean up all data objects and free up all resources.*
- [getv](#) ()  
*Method to get our value (lever position).*
- [setv](#) (state)  
*Method to set out value (level position).*
- [geti](#) (ind)  
*Method to get the state of one of out indicators.*
- [seti](#) (ind, value)  
*Method to set an indicator's state.*
- [invoke](#) ()  
*Method to invoke the switch plate.*

## Private Member Functions

- [\\_configureLabel](#) (option, value)  
*Method to update the label option.*

## Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*

## Static Private Attributes

- static [\\_PlatePolygon](#)  
*Polygon coordinates for the plate.*

### 9.141.1 Detailed Description

Signal plate object type.

These are on the control panel and represent levers for controlling track signals (control point signals). They have a lever that can be in three positions, Left, Center, or Right.

#### Parameters

<a href="#">_ctcpanel</a>	The <a href="#">CTCPanel</a> megawidget.
<a href="#">_canvas</a>	The control panel canvas to draw the switch plate on.
...	Options: <ul style="list-style-type: none"> <li>• -x The x coordinate of the object (readonly, default 0).</li> <li>• -y The y coordinate of the object (readonly, default 0).</li> <li>• -label The label of the switch plate (default 1).</li> <li>• -controlpoint The name of the control point this switch is part of (readonly, default CP1).</li> <li>• -leftcommand The Tcl script to run when switch is set to left (default {}).</li> <li>• -centercommand The Tcl script to run when switch is set to center (default {}).</li> <li>• -rightcommand The Tcl script to run when switch is set to right (default {}).</li> </ul>

Defined coords terminals:

- xy The base coords of the object.

Defined values (states):

- Left Left position.
- Right Right position.
- Center Center position.

Defined indicators:

- L Left indicator, green if on.
- C Center indicator, red if on.
- R Right indicator, green if on.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.141.2 Constructor & Destructor Documentation

#### 9.141.2.1 SIGPlate()

```
CTCPanel::SIGPlate::SIGPlate (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a [SIGPlate](#) object.

#### Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The control panel canvas to draw the switch plate on.
<code>...</code>	Option list.

#### 9.141.2.2 ~SIGPlate()

```
CTCPanel::SIGPlate::~~SIGPlate ( )
```

Clean up all data objects and free up all resources.

### 9.141.3 Member Function Documentation

#### 9.141.3.1 `_configureLabel()`

```
CTCPanel::SIGPlate::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

#### 9.141.3.2 `geti()`

```
CTCPanel::SIGPlate::geti (
    ind )
```

Method to get the state of one of out indicators.

##### Parameters

<i>ind</i>	The indicator to fetch state information for.
------------	---

#### 9.141.3.3 `getv()`

```
CTCPanel::SIGPlate::getv ( )
```

Method to get our value (lever position).

#### 9.141.3.4 `invoke()`

```
CTCPanel::SIGPlate::invoke ( )
```

Method to invoke the switch plate.

One of the command scripts is executed depending on the lever position.

#### 9.141.3.5 `seti()`

```
CTCPanel::SIGPlate::seti (
    ind ,
    value )
```

Method to set an indicator's state.

**Parameters**

<i>ind</i>	The indicator to set.
<i>value</i>	The state to set it to.

**9.141.3.6 setv()**

```
CTCPanel::SIGPlate::setv (  
    state )
```

Method to set out value (level position).

**Parameters**

<i>state</i>	The new state to set.
--------------	-----------------------

**9.141.4 Member Data Documentation****9.141.4.1 \_PlatePolygon**

```
CTCPanel::SIGPlate::_PlatePolygon [static], [private]
```

Polygon coordinates for the plate.

**9.141.4.2 canvas**

```
CTCPanel::SIGPlate::canvas [private]
```

The canvas component (parent widget component).

**9.141.4.3 ctcpnl**

```
CTCPanel::SIGPlate::ctcpnl [private]
```

The CTC Panel component (parent widget).

## 9.142 SimpleDOMElement Class Reference

A simple DOM element coded in Tcl using SNIT.

### Public Member Functions

- [SimpleDOMElement](#) (name,...)  
*The constructor.*
- [children](#) ()  
*Method to return the elements children.*
- [addchild](#) (childnode)  
*Method to add a child node.*
- [length](#) ()  
*Method to return the number of children.*
- [data](#) ()  
*Method to return the element's data.*
- [setdata](#) (d)  
*Method to set the element's data.*
- [display](#) (fp="stdout", indent="")  
*Method to display a node, along with its children, and a proper XML document.*
- [attribute](#) (attrname)  
*Method to return a selected attribute's value.*
- [setAttribute](#) (attrname, value="")  
*Method to set a selected attribute's value.*
- [getElementsByTagName](#) (thetag,...)  
*Method to return all of the elements under this element with the specified tag name.*
- [getElementsById](#) (theid)  
*Method to return all of the elements under this element with the specified value of their id attribute.*
- [isChild](#) (item)  
*Method to check if the item is a child of this node.*
- [getParent](#) (item)  
*Method to get the parent of the item.*
- [removeChild](#) (item)  
*Method to remove item from the children of this node.*

### Static Public Member Functions

- static [\\_formattrlist](#) (attrs)  
*Format a attribute list for inclusion in displayed XML.*
- static [\\_quoteXML](#) (text)  
*Escape text for inclusion in displayed XML.*
- static [validate](#) (object)  
*Validation typemethod.*



## Private Attributes

- [\\_data](#)  
*The element's data.*
- [\\_children](#)  
*The element's children.*

### 9.142.1 Detailed Description

A simple DOM element coded in Tcl using SNIT.

This class implements a simplified DOM element, that implements the `getElementsByTagName` and `getElementsById` methods, along with accessors to get data, attributes, and children of XML elements.

#### Parameters

<i>name</i>	Element name. Generally %%AUTO%% is passed.
—	Options: <ul style="list-style-type: none"><li>• <code>-tag</code> The element's tag.</li><li>• <code>-attributes</code> The element's attributes.</li><li>• <code>-opts</code> The element's options.</li></ul>

#### Author

Robert Heller <heller@deepsoft.com>.

### 9.142.2 Constructor & Destructor Documentation

#### 9.142.2.1 SimpleDOMElement()

```
SimpleDOMElement::SimpleDOMElement (  
    name ,  
    ... )
```

The constructor.

Just sets the options.

### 9.142.3 Member Function Documentation

#### 9.142.3.1 `_formattrlist()`

```
static SimpleDOMElement::_formattrlist (
    attrs ) [static]
```

Format a attribute list for inclusion in displayed XML.

##### Parameters

<i>attrs</i>	The attribute list as a alternating list of names and values.
--------------	---

##### Returns

A formatted and escaped attribute list string.

#### 9.142.3.2 `_quoteXML()`

```
static SimpleDOMElement::_quoteXML (
    text ) [static]
```

Escape text for inclusion in displayed XML.

##### Parameters

<i>text</i>	Unescaped string.
-------------	-------------------

##### Returns

A properly escaped XML string.

#### 9.142.3.3 `addchild()`

```
SimpleDOMElement::addchild (
    childnode )
```

Method to add a child node.

##### Parameters

<i>childnode</i>	The child node to add.
------------------	------------------------

#### 9.142.3.4 attribute()

```
SimpleDOMElement::attribute (
    attrname )
```

Method to return a selected attribute's value.

##### Parameters

<i>attrname</i>	The name of the attribute.
-----------------	----------------------------

##### Returns

The attribute's value or the empty string.

#### 9.142.3.5 children()

```
SimpleDOMElement::children ( )
```

Method to return the elements children.

##### Returns

The children.

#### 9.142.3.6 data()

```
SimpleDOMElement::data ( )
```

Method to return the element's data.

##### Returns

The data.

#### 9.142.3.7 display()

```
SimpleDOMElement::display (
    fp = "stdout",
    indent = "" )
```

Method to display a node, along with its children, and a proper XML document.

**Parameters**

<i>fp</i>	Channel to write the display to.
<i>indent</i>	The indentation to use.

**9.142.3.8 getElementById()**

```
SimpleDOMElement::getElementById (
    theid )
```

Method to return all of the elements under this element with the specified value of their id attribute.

**Parameters**

<i>theid</i>	The id value match.
--------------	---------------------

**Returns**

A list of element object with the matching id value.

**9.142.3.9 getElementByTagName()**

```
SimpleDOMElement::getElementByTagName (
    thetag ,
    ... )
```

Method to return all of the elements under this element with the specified tag name.

**Parameters**

<i>thetag</i>	The tag to match.
---------------	-------------------

**Returns**

A list of element object with the matching tag.

#### 9.142.3.10 getParent()

```
SimpleDOMElement::getParent (
    item )
```

Method to get the parent of the item.

##### Parameters

<i>item</i>	The item to get the parent of.
-------------	--------------------------------

##### Returns

The parent node or {} if none found.

#### 9.142.3.11 isChild()

```
SimpleDOMElement::isChild (
    item )
```

Method to check if the item is a child of this node.

##### Parameters

<i>item</i>	The possible child.
-------------	---------------------

##### Returns

True if item is a child, false otherwise.

#### 9.142.3.12 length()

```
SimpleDOMElement::length ( )
```

Method to return the number of children.

##### Returns

The number of children.

**9.142.3.13 removeChild()**

```
SimpleDOMElement::removeChild (
    item )
```

Method to remove item from the children of this node.

**Parameters**

<i>item</i>	The item to remove.
-------------	---------------------

**9.142.3.14 setAttribute()**

```
SimpleDOMElement::setAttribute (
    attrname ,
    value = "" )
```

Method to set a selected attribute's value.

**Parameters**

<i>attrname</i>	The name of the attribute.
<i>value</i>	The value to set. Default is the empty string.

**9.142.3.15 setdata()**

```
SimpleDOMElement::setdata (
    d )
```

Method to set the element's data.

**Parameters**

<i>d</i>	The new data.
----------	---------------

**9.142.3.16 validate()**

```
static SimpleDOMElement::validate (
    object ) [static]
```

Validation type method.

Raises an error if its argument is not a [SimpleDOMElement](#) object.

#### Parameters

<i>object</i>	The object to typecheck.
---------------	--------------------------

#### Returns

The object or raise an error.

### 9.142.4 Member Data Documentation

#### 9.142.4.1 \_children

`SimpleDOMElement::_children` [private]

The element's children.

#### 9.142.4.2 \_data

`SimpleDOMElement::_data` [private]

The element's data.

## 9.143 CTCPanel::SingleSlip Class Reference

Single Slip (turnout) object type.

### Public Member Functions

- [SingleSlip](#) (name, \_ctcpanel, \_canvas,...)  
*Construct a [SingleSlip](#) object.*
- [~SingleSlip](#) ()  
*Clean up all data objects and free up all resources.*
- [getv](#) ()  
*Method to get our value (state).*
- [setv](#) (value)  
*Method to set out value (state).*
- [geti](#) (ind)  
*Method to get the state of one of our indicators (none).*
- [seti](#) (ind, value)  
*Method to set an indicator's state (none).*
- [invoke](#) ()  
*Method to invoke the switch.*

## Private Member Functions

- [\\_configureLabel](#) (option, value)  
*Method to update the label option.*

## Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*
- [state](#)  
*The state of the points.*

### 9.143.1 Detailed Description

Single Slip (turnout) object type.

These are on the schematic and represent a switch on the Schematic.

#### Parameters

<a href="#">_ctcpanel</a>	The <a href="#">CTCPanel</a> megawidget.
<a href="#">_canvas</a>	The schematic canvas to draw the switch on.
...	Options: <ul style="list-style-type: none"> <li>• -x The x coordinate of the object (readonly, default 0).</li> <li>• -y The y coordinate of the object (readonly, default 0).</li> <li>• -controlpoint The name of the control point this label is part of (readonly, default CP1).</li> <li>• -label The label of the switch (default "1").</li> <li>• -orientation The orientation (8-way) of the switch (readonly, default 0).</li> <li>• -flipped Whether or not the switch is flipped (readonly, default no).</li> <li>• -statecommand A command to run to get the switch's state (default {}).</li> <li>• -occupiedcommand A command to run to find out if the switch is occupied (default {}).</li> </ul>

Defined coords terminals:

- MainL Mainline left.
- MainR Mainline right.
- AltL Alternative line left.
- AltR Alternative line right.



Defined values (states):

- Normal Points are aligned for the mainline.
- Reverse Points are aligned for the branchline.
- Unknown Point are not aligned for any route (eg the points are in motion).

Defined indicators: none.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.143.2 Constructor & Destructor Documentation

#### 9.143.2.1 SingleSlip()

```
CTCPanel::SingleSlip::SingleSlip (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a [SingleSlip](#) object.

#### Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the <a href="#">SingleSlip</a> on.
<code>...</code>	Option list.

#### 9.143.2.2 ~SingleSlip()

```
CTCPanel::SingleSlip::~~SingleSlip ( )
```

Clean up all data objects and free up all resources.

### 9.143.3 Member Function Documentation

**9.143.3.1 \_configureLabel()**

```
CTCPanel::SingleSlip::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

**9.143.3.2 geti()**

```
CTCPanel::SingleSlip::geti (
    ind )
```

Method to get the state of one of our indicators (none).

**9.143.3.3 getv()**

```
CTCPanel::SingleSlip::getv ( )
```

Method to get our value (state).

**9.143.3.4 invoke()**

```
CTCPanel::SingleSlip::invoke ( )
```

Method to invoke the switch.

**9.143.3.5 seti()**

```
CTCPanel::SingleSlip::seti (
    ind ,
    value )
```

Method to set an indicator's state (none).

**9.143.3.6 setv()**

```
CTCPanel::SingleSlip::setv (
    value )
```

Method to set out value (state).

## Parameters

<i>value</i>	The new state to set.
--------------	-----------------------

## 9.143.4 Member Data Documentation

## 9.143.4.1 canvas

CTCPanel::SingleSlip::canvas [private]

The canvas component (parent widget component).

## 9.143.4.2 ctcpnl

CTCPanel::SingleSlip::ctcpnl [private]

The CTC Panel component (parent widget).

## 9.143.4.3 state

CTCPanel::SingleSlip::state [private]

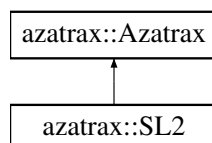
The state of the points.

## 9.144 azatrax::SL2 Class Reference

[SL2](#) I/O Class.

```
#include <sl2.h>
```

Inheritance diagram for azatrax::SL2:



## Classes

- union [status1\\_union](#)  
*Status byte 1 union type (Output states)*
- union [status2\\_union](#)  
*Status byte 2 union type (Input sense)*
- union [status3\\_union](#)  
*Status byte 3 union type (Input control state)*

## Public Member Functions

- [~SL2](#) ()
- [ErrorCode SetQ1posQ2neg](#) () const  
*Sets output terminal Q1 to positive, Q2 to negative.*
- [ErrorCode SetQ1negQ2pos](#) () const  
*Sets output terminal Q1 to negative, Q2 to positive.*
- [ErrorCode SetQ1Q2open](#) () const  
*Outputs Q1 & Q2 both set to open circuit (disconnects switch machine #1)*
- [ErrorCode SetQ3posQ4neg](#) () const  
*Sets output terminal Q3 to positive, Q4 to negative.*
- [ErrorCode SetQ3negQ4pos](#) () const  
*Sets output terminal Q3 to negative, Q4 to positive.*
- [ErrorCode SetQ3Q4open](#) () const  
*Outputs Q3 & Q4 both set to open circuit (disconnects switch machine #2)*
- [ErrorCode OutputRelayInputControl](#) (bool I1, bool I2, bool I3, bool I4)  
*Enable/Disable discrete input lines from affecting outputs.*
- bool [Sense\\_1](#) () const  
*Sense 1, return true if input line 1 was activated since last get status.*
- bool [Sense\\_2](#) () const  
*Sense 2, return true if input line 2 was activated since last get status.*
- bool [Sense\\_3](#) () const  
*Sense 3, return true if input line 3 was activated since last get status.*
- bool [Sense\\_4](#) () const  
*Sense 4, return true if input line 4 was activated since last get status.*
- bool [Motor\\_1\\_Direction](#) () const  
*Motor 1 direction, return true if Q1 is positive.*
- bool [Motor\\_1\\_State](#) () const  
*Motor 1 state, return true if Q1 and Q2 are on.*
- bool [Motor\\_2\\_Direction](#) () const  
*Motor 2 direction, return true if Q3 is positive.*
- bool [Motor\\_2\\_State](#) () const  
*Motor 2 state, return true if Q3 and Q4 are on.*
- bool [Input\\_1\\_Enabled](#) () const  
*Input 1 enabled? Return true if I1 can affect outputs.*
- bool [Input\\_2\\_Enabled](#) () const  
*Input 2 enabled? Return true if I2 can affect outputs.*
- bool [Input\\_3\\_Enabled](#) () const  
*Input 3 enabled? Return true if I3 can affect outputs.*
- bool [Input\\_4\\_Enabled](#) () const  
*Input 4 enabled? Return true if I4 can affect outputs.*

## Private Member Functions

- [SL2](#) (const char \*serialnumber, char \*\*outmessage=NULL)  
*Base constructor.*

## Friends

- class [Azatrax](#)

## Additional Inherited Members

### 9.144.1 Detailed Description

[SL2](#) I/O Class.

[SL2](#) interface class.

This class implements the interface logic for a SL2-U device.

The constructor opens a connection to a SL2-U device, given its serial number. Each SL2-U device has a unique, factory defined serial number, which is printed on a sticker attached to the module. This serial number is much like the MAC address of an Ethernet interface. The destructor closes the connection to the device and frees any resources allocated.

The class provides methods to send commands to the device, read back its state and interrogate the state read back. This way each class instance encapsulates each device instance.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.144.2 Constructor & Destructor Documentation

#### 9.144.2.1 SL2()

```
azatrax::SL2::SL2 (
    const char * serialnumber,
    char ** outmessage = NULL ) [inline], [private]
```

Base constructor.

#### Parameters

<i>serialnumber</i>	The serial number of the device to open.
<i>outmessage</i>	To hold an error message, if any.

### 9.144.2.2 ~SL2()

```
azatrax::SL2::~~SL2 ( ) [inline]
```

## 9.144.3 Member Function Documentation

### 9.144.3.1 Input\_1\_Enabled()

```
bool azatrax::SL2::Input_1_Enabled ( ) const [inline]
```

Input 1 enabled? Return true if I1 can affect outputs.

References azatrax::SL2::status3\_union::input\_1\_enabled, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::↵  
StateDataPacket::status3, azatrax::SL2::status3\_union::theBits, and azatrax::SL2::status3\_union::theByte.

### 9.144.3.2 Input\_2\_Enabled()

```
bool azatrax::SL2::Input_2_Enabled ( ) const [inline]
```

Input 2 enabled? Return true if I2 can affect outputs.

References azatrax::SL2::status3\_union::input\_2\_enabled, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::↵  
StateDataPacket::status3, azatrax::SL2::status3\_union::theBits, and azatrax::SL2::status3\_union::theByte.

### 9.144.3.3 Input\_3\_Enabled()

```
bool azatrax::SL2::Input_3_Enabled ( ) const [inline]
```

Input 3 enabled? Return true if I3 can affect outputs.

References azatrax::SL2::status3\_union::input\_3\_enabled, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::↵  
StateDataPacket::status3, azatrax::SL2::status3\_union::theBits, and azatrax::SL2::status3\_union::theByte.

#### 9.144.3.4 Input\_4\_Enabled()

```
bool azatrax::SL2::Input_4_Enabled ( ) const [inline]
```

Input 4 enabled? Return true if I4 can affect outputs.

References azatrax::SL2::status3\_union::input\_4\_enabled, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::↵StateDataPacket::status3, azatrax::SL2::status3\_union::theBits, and azatrax::SL2::status3\_union::theByte.

#### 9.144.3.5 Motor\_1\_Direction()

```
bool azatrax::SL2::Motor_1_Direction ( ) const [inline]
```

Motor 1 direction, return true if Q1 is positive.

References azatrax::SL2::status1\_union::motor\_1\_direction, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::↵StateDataPacket::status1, azatrax::SL2::status1\_union::theBits, and azatrax::SL2::status1\_union::theByte.

#### 9.144.3.6 Motor\_1\_State()

```
bool azatrax::SL2::Motor_1_State ( ) const [inline]
```

Motor 1 state, return true if Q1 and Q2 are on.

References azatrax::SL2::status1\_union::motor\_1\_state, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::State↵DataPacket::status1, azatrax::SL2::status1\_union::theBits, and azatrax::SL2::status1\_union::theByte.

#### 9.144.3.7 Motor\_2\_Direction()

```
bool azatrax::SL2::Motor_2_Direction ( ) const [inline]
```

Motor 2 direction, return true if Q3 is positive.

References azatrax::SL2::status1\_union::motor\_2\_direction, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::↵StateDataPacket::status1, azatrax::SL2::status1\_union::theBits, and azatrax::SL2::status1\_union::theByte.

#### 9.144.3.8 Motor\_2\_State()

```
bool azatrax::SL2::Motor_2_State ( ) const [inline]
```

Motor 2 state, return true if Q3 and Q4 are on.

References azatrax::SL2::status1\_union::motor\_2\_state, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateDataPacket::status1, azatrax::SL2::status1\_union::theBits, and azatrax::SL2::status1\_union::theByte.

#### 9.144.3.9 OutputRelayInputControl()

```
ErrorCode azatrax::SL2::OutputRelayInputControl (
    bool I1,
    bool I2,
    bool I3,
    bool I4 ) [inline]
```

Enable/Disable discrete input lines from affecting outputs.

When enabled, I1 & I2 affect Q1 & Q2 (switch 1), I3 & I4 affect Q3 & Q4 (switch 2).

##### Parameters

<i>I1</i>	Enable/Disable I1.
<i>I2</i>	Enable/Disable I2.
<i>I3</i>	Enable/Disable I3.
<i>I4</i>	Enable/Disable I4.

References azatrax::Azatrax::cmd\_OutputRelayInputControl, and azatrax::Azatrax::send2Bytes().

#### 9.144.3.10 Sense\_1()

```
bool azatrax::SL2::Sense_1 ( ) const [inline]
```

Sense 1, return true if input line 1 was activated since last get status.

References azatrax::SL2::status2\_union::sense\_1, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateDataPacket::status2, azatrax::SL2::status2\_union::theBits, and azatrax::SL2::status2\_union::theByte.



#### 9.144.3.11 Sense\_2()

```
bool azatrax::SL2::Sense_2 ( ) const [inline]
```

Sense 2, return true if input line 2 was activated since last get status.

References azatrax::SL2::status2\_union::sense\_2, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateData↔Packet::status2, azatrax::SL2::status2\_union::theBits, and azatrax::SL2::status2\_union::theByte.

#### 9.144.3.12 Sense\_3()

```
bool azatrax::SL2::Sense_3 ( ) const [inline]
```

Sense 3, return true if input line 3 was activated since last get status.

References azatrax::SL2::status2\_union::sense\_3, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateData↔Packet::status2, azatrax::SL2::status2\_union::theBits, and azatrax::SL2::status2\_union::theByte.

#### 9.144.3.13 Sense\_4()

```
bool azatrax::SL2::Sense_4 ( ) const [inline]
```

Sense 4, return true if input line 4 was activated since last get status.

References azatrax::SL2::status2\_union::sense\_4, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateData↔Packet::status2, azatrax::SL2::status2\_union::theBits, and azatrax::SL2::status2\_union::theByte.

#### 9.144.3.14 SetQ1negQ2pos()

```
ErrorCode azatrax::SL2::SetQ1negQ2pos ( ) const [inline]
```

Sets output terminal Q1 to negative, Q2 to positive.

References azatrax::Azatrax::cmd\_Q1negQ2pos, and azatrax::Azatrax::sendByte().

#### 9.144.3.15 SetQ1posQ2neg()

```
ErrorCode azatrax::SL2::SetQ1posQ2neg ( ) const [inline]
```

Sets output terminal Q1 to positive, Q2 to negative.

References azatrax::Azatrax::cmd\_Q1posQ2neg, and azatrax::Azatrax::sendByte().

#### 9.144.3.16 SetQ1Q2open()

```
ErrorCode azatrax::SL2::SetQ1Q2open ( ) const [inline]
```

Outputs Q1 & Q2 both set to open circuit (disconnects switch machine #1)

References azatrax::Azatrax::cmd\_Q1Q2open, and azatrax::Azatrax::sendByte().

#### 9.144.3.17 SetQ3negQ4pos()

```
ErrorCode azatrax::SL2::SetQ3negQ4pos ( ) const [inline]
```

Sets output terminal Q3 to negative, Q4 to positive.

References azatrax::Azatrax::cmd\_Q3negQ4pos, and azatrax::Azatrax::sendByte().

#### 9.144.3.18 SetQ3posQ4neg()

```
ErrorCode azatrax::SL2::SetQ3posQ4neg ( ) const [inline]
```

Sets output terminal Q3 to positive, Q4 to negative.

References azatrax::Azatrax::cmd\_Q3posQ4neg, and azatrax::Azatrax::sendByte().

#### 9.144.3.19 SetQ3Q4open()

```
ErrorCode azatrax::SL2::SetQ3Q4open ( ) const [inline]
```

Outputs Q3 & Q4 both set to open circuit (disconnects switch machine #2)

References azatrax::Azatrax::cmd\_Q3Q4open, and azatrax::Azatrax::sendByte().

#### 9.144.4 Friends And Related Function Documentation

##### 9.144.4.1 Azatrax

```
friend class Azatrax [friend]
```

### 9.145 xpressnet::SoftwareVersion Class Reference

Software version.

#### Public Member Functions

- [SoftwareVersion](#) (name, majornibble, minornibble, cst=0xff)  
*Constructor.*
- [Major](#) ()  
*Return major version number.*
- [Minor](#) ()  
*Return minor version number.*
- [CommandStationTypeCode](#) ()  
*Return command station type.*

#### Private Attributes

- [\\_major](#)  
*Major version number.*
- [\\_minor](#)  
*Minor version number.*
- [\\_command\\_station\\_type](#)  
*Command station type.*

#### 9.145.1 Detailed Description

Software version.

#### Author

Robert Heller <heller@deepsoft.com>

## 9.145.2 Constructor & Destructor Documentation

### 9.145.2.1 SoftwareVersion()

```
xpressnet::SoftwareVersion::SoftwareVersion (
    name ,
    majornibble ,
    minornibble ,
    cst = 0xff )
```

Constructor.

#### Parameters

<i>majornibble</i>	Major version number.
<i>minornibble</i>	Minor version number.
<i>cst</i>	Command station type.

## 9.145.3 Member Function Documentation

### 9.145.3.1 CommandStationTypeCode()

```
xpressnet::SoftwareVersion::CommandStationTypeCode ( )
```

Return command station type.

### 9.145.3.2 Major()

```
xpressnet::SoftwareVersion::Major ( )
```

Return major version number.

### 9.145.3.3 Minor()

```
xpressnet::SoftwareVersion::Minor ( )
```

Return minor version number.

## 9.145.4 Member Data Documentation

### 9.145.4.1 `_command_station_type`

`xpressnet::SoftwareVersion::_command_station_type` [private]

Command station type.

### 9.145.4.2 `_major`

`xpressnet::SoftwareVersion::_major` [private]

Major version number.

### 9.145.4.3 `_minor`

`xpressnet::SoftwareVersion::_minor` [private]

Minor version number.

## 9.146 splash Class Reference

Widget that implements a splash window.

### Public Member Functions

- [update](#) (statusMessage, percentDone)  
*Method to update the splash window.*
- [enableClickDestroy](#) ()  
*Method to enable click to destroy.*
- [hide](#) ()  
*Method to hide the splash window.*
- [show](#) ()  
*Method to show the splash window.*
- [splash](#) (name,...)  
*Constructor initialize a splash window.*

## Private Member Functions

- [CheckColor](#) (option, value)  
*Method to validate a color option.*
- [CheckImage](#) (option, value)  
*Method to validate an image option.*

## Private Attributes

- [image](#)  
*Image component.*
- [progressBar](#)  
*Progress bar component.*
- [title](#)  
*Title component.*
- [icon](#)  
*Icon component.*
- [status](#)  
*Status component.*
- [header](#)  
*Header component.*
- [currentProgress](#)  
*The current progress.*

### 9.146.1 Detailed Description

Widget that implements a splash window.

A splash window is a toplevel that is displayed during startup and shows a startup graphic and shows the startup / initialization progress.

#### Parameters

<i>path</i>	The widget path.
...	Options: <ul style="list-style-type: none"><li>• -style Style name, default is Splash.</li><li>• -titleforeground Delegated to the title widget as -foreground.</li><li>• -statusforeground Delegated to the status widget as -foreground.</li><li>• -background Background color.</li><li>• -progressbar Flag that enables or disables the progress bar.</li><li>• -image Spash image to display in the lower part of the splash window.</li><li>• -icon Icon to display next to the text in the upper part of the splash window.</li><li>• -title Title text.</li></ul>

**Author**

Robert Heller <heller@deepsoft.com>

**9.146.2 Package provided**

Splash 1.0

**9.146.3 Constructor & Destructor Documentation****9.146.3.1 splash()**

```
splash::splash (
    name ,
    ... )
```

Constructor initialize a splash window.

**Parameters**

...	Option list. [index] constructor!splash
-----	---

**9.146.4 Member Function Documentation****9.146.4.1 CheckColor()**

```
splash::CheckColor (
    option ,
    value ) [private]
```

Method to validate a color option.

**Parameters**

<i>option</i>	The option being set.
<i>value</i>	The value it is being set to.

#### 9.146.4.2 CheckImage()

```
splash::CheckImage (
    option ,
    value ) [private]
```

Method to validate an image option.

##### Parameters

<i>option</i>	The option being set.
<i>value</i>	The value it is being set to.

#### 9.146.4.3 enableClickDestroy()

```
splash::enableClickDestroy ( )
```

Method to enable click to destroy.

#### 9.146.4.4 hide()

```
splash::hide ( )
```

Method to hide the splash window.

#### 9.146.4.5 show()

```
splash::show ( )
```

Method to show the splash window.

#### 9.146.4.6 update()

```
splash::update (
    statusMessage ,
    percentDone )
```

Method to update the splash window.



## Parameters

<i>statusMessage</i>	The new status message.
<i>percentDone</i>	The percent completed.

## 9.146.5 Member Data Documentation

### 9.146.5.1 currentProgress

`splash::currentProgress` [private]

The current progress.

### 9.146.5.2 header

`splash::header` [private]

Header component.

### 9.146.5.3 icon

`splash::icon` [private]

Icon component.

### 9.146.5.4 image

`splash::image` [private]

Image component.

#### 9.146.5.5 progressBar

```
splash::progressBar [private]
```

Progress bar component.

#### 9.146.5.6 status

```
splash::status [private]
```

Status component.

#### 9.146.5.7 title

```
splash::title [private]
```

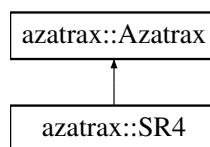
Title component.

### 9.147 azatrax::SR4 Class Reference

[SR4](#) I/O Class.

```
#include <sr4.h>
```

Inheritance diagram for azatrax::SR4:



#### Classes

- union [status1\\_union](#)  
*Status byte 1 union type (Outputs)*
- union [status2\\_union](#)  
*Status byte 2 union type (Input sense)*
- union [status3\\_union](#)  
*Status byte 3 union type (Input Control Status)*

## Public Member Functions

- [~SR4](#) ()
- [ErrorCode BlinkRelays](#) (bool Q1, bool Q2, bool Q3, bool Q4, uint8\_t speed)  
*Blink relay contacts.*
- [ErrorCode RelaysOff](#) (bool Q1, bool Q2, bool Q3, bool Q4)  
*Set output relay contacts off.*
- [ErrorCode RelaysOn](#) (bool Q1, bool Q2, bool Q3, bool Q4)  
*Set output relay contacts on.*
- [ErrorCode PulseRelays](#) (bool Q1, bool Q2, bool Q3, bool Q4, uint8\_t duration)  
*Pulse output relay contacts.*
- [ErrorCode OutputRelayInputControl](#) (bool I1, bool I2, bool I3, bool I4)  
*Enable/Disable discrete input lines from affecting outputs.*
- bool [Sense\\_1\\_Latch](#) () const  
*Sense 1, return true if input line 1 was activated since last get status.*
- bool [Sense\\_2\\_Latch](#) () const  
*Sense 2, return true if input line 2 was activated since last get status.*
- bool [Sense\\_3\\_Latch](#) () const  
*Sense 3, return true if input line 3 was activated since last get status.*
- bool [Sense\\_4\\_Latch](#) () const  
*Sense 4, return true if input line 4 was activated since last get status.*
- bool [Q1\\_State](#) () const  
*Q1 state, return true if Q1 is closed.*
- bool [Q2\\_State](#) () const  
*Q2 state, return true if Q2 is closed.*
- bool [Q3\\_State](#) () const  
*Q3 state, return true if Q3 is closed.*
- bool [Q4\\_State](#) () const  
*Q4 state, return true if Q4 is closed.*
- bool [Input\\_1\\_Enabled](#) () const  
*Input 1 enabled? Return true if I1 can affect outputs.*
- bool [Input\\_2\\_Enabled](#) () const  
*Input 2 enabled? Return true if I2 can affect outputs.*
- bool [Input\\_3\\_Enabled](#) () const  
*Input 3 enabled? Return true if I3 can affect outputs.*
- bool [Input\\_4\\_Enabled](#) () const  
*Input 4 enabled? Return true if I4 can affect outputs.*
- bool [Sense\\_1\\_Live](#) () const  
*Sense 1, return true if input line 1 is now activated.*
- bool [Sense\\_2\\_Live](#) () const  
*Sense 2, return true if input line 2 is now activated.*
- bool [Sense\\_3\\_Live](#) () const  
*Sense 3, return true if input line 3 is now activated.*
- bool [Sense\\_4\\_Live](#) () const  
*Sense 4, return true if input line 4 is now activated.*

## Private Member Functions

- [SR4](#) (const char \*serialnumber, char \*\*outmessage=NULL)  
*Base constructor.*

## Friends

- class [Azatrax](#)

## Additional Inherited Members

### 9.147.1 Detailed Description

[SR4](#) I/O Class.

[SR4](#) interface class.

This class implements the interface logic for a SR4-U device.

The constructor opens a connection to a SR4-U device, given its serial number. Each SR4-U device has a unique, factory defined serial number, which is printed on a sticker attached to the module. This serial number is much like the MAC address of an Ethernet interface. The destructor closes the connection to the device and frees any resources allocated.

The class provides methods to send commands to the device, read back its state and interrogate the state read back. This way each class instance encapsulates each device instance.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.147.2 Constructor & Destructor Documentation

#### 9.147.2.1 SR4()

```
azatrax::SR4::SR4 (
    const char * serialnumber,
    char ** outmessage = NULL ) [inline], [private]
```

Base constructor.

## Parameters

<i>serialnumber</i>	The serial number of the device to open.
<i>outmessage</i>	To hold an error message, if any.

## 9.147.2.2 ~SR4()

```
azatrax::SR4::~~SR4 ( ) [inline]
```

## 9.147.3 Member Function Documentation

## 9.147.3.1 BlinkRelays()

```
ErrorCode azatrax::SR4::BlinkRelays (
    bool Q1,
    bool Q2,
    bool Q3,
    bool Q4,
    uint8_t speed ) [inline]
```

Blink relay contacts.

Sets output relay contacts to blinking state.

## Parameters

<i>Q1</i>	Blink relay Q1.
<i>Q2</i>	Blink relay Q2.
<i>Q3</i>	Blink relay Q3.
<i>Q4</i>	Blink relay Q4.
<i>speed</i>	Blink speed: 0 is 4hz, 1 is 2hz, 2 is 1hz, and 3 is .5hz.

References `azatrax::Azatrax::cmd_OutputRelayBlink`, and `azatrax::Azatrax::send3Bytes()`.

## 9.147.3.2 Input\_1\_Enabled()

```
bool azatrax::SR4::Input_1_Enabled ( ) const [inline]
```

Input 1 enabled? Return true if I1 can affect outputs.

References azatrax::SR4::status3\_union::input\_1\_enabled, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::↵  
StateDataPacket::status3, azatrax::SR4::status3\_union::theBits, and azatrax::SR4::status3\_union::theByte.

#### 9.147.3.3 Input\_2\_Enabled()

```
bool azatrax::SR4::Input_2_Enabled ( ) const [inline]
```

Input 2 enabled? Return true if I2 can affect outputs.

References azatrax::SR4::status3\_union::input\_2\_enabled, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::↵  
StateDataPacket::status3, azatrax::SR4::status3\_union::theBits, and azatrax::SR4::status3\_union::theByte.

#### 9.147.3.4 Input\_3\_Enabled()

```
bool azatrax::SR4::Input_3_Enabled ( ) const [inline]
```

Input 3 enabled? Return true if I3 can affect outputs.

References azatrax::SR4::status3\_union::input\_3\_enabled, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::↵  
StateDataPacket::status3, azatrax::SR4::status3\_union::theBits, and azatrax::SR4::status3\_union::theByte.

#### 9.147.3.5 Input\_4\_Enabled()

```
bool azatrax::SR4::Input_4_Enabled ( ) const [inline]
```

Input 4 enabled? Return true if I4 can affect outputs.

References azatrax::SR4::status3\_union::input\_4\_enabled, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::↵  
StateDataPacket::status3, azatrax::SR4::status3\_union::theBits, and azatrax::SR4::status3\_union::theByte.

#### 9.147.3.6 OutputRelayInputControl()

```
ErrorCode azatrax::SR4::OutputRelayInputControl (
    bool I1,
    bool I2,
    bool I3,
    bool I4 ) [inline]
```

Enable/Disable discrete input lines from affecting outputs.

When enabled, I1 & I2 affect Q1 & Q2 (switch 1), I3 & I4 affect Q3 & Q4 (switch 2).

## Parameters

<i>I1</i>	Enable/Disable I1.
<i>I2</i>	Enable/Disable I2.
<i>I3</i>	Enable/Disable I3.
<i>I4</i>	Enable/Disable I4.

References `azatrax::Azatrax::cmd_OutputRelayInputControl`, and `azatrax::Azatrax::send2Bytes()`.

## 9.147.3.7 PulseRelays()

```

ErrorCode azatrax::SR4::PulseRelays (
    bool Q1,
    bool Q2,
    bool Q3,
    bool Q4,
    uint8_t duration ) [inline]

```

Pulse output relay contacts.

## Parameters

<i>Q1</i>	Pulse Q1.
<i>Q2</i>	Pulse Q2.
<i>Q3</i>	Pulse Q3.
<i>Q4</i>	Pulse Q4.
<i>duration</i>	Pulse duration in 0.5 second units.

References `azatrax::Azatrax::cmd_OutputRelayPulse`, and `azatrax::Azatrax::send3Bytes()`.

## 9.147.3.8 Q1\_State()

```
bool azatrax::SR4::Q1_State ( ) const [inline]
```

Q1 state, return true if Q1 is closed.

References `azatrax::SR4::status1_union::Q1_state`, `azatrax::Azatrax::stateDataPacket`, `azatrax::Azatrax::StateDataPacket::status1`, `azatrax::SR4::status1_union::theBits`, and `azatrax::SR4::status1_union::theByte`.

**9.147.3.9 Q2\_State()**

```
bool azatrax::SR4::Q2_State ( ) const [inline]
```

Q2 state, return true if Q2 is closed.

References azatrax::SR4::status1\_union::Q2\_state, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateData↔Packet::status1, azatrax::SR4::status1\_union::theBits, and azatrax::SR4::status1\_union::theByte.

**9.147.3.10 Q3\_State()**

```
bool azatrax::SR4::Q3_State ( ) const [inline]
```

Q3 state, return true if Q3 is closed.

References azatrax::SR4::status1\_union::Q3\_state, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateData↔Packet::status1, azatrax::SR4::status1\_union::theBits, and azatrax::SR4::status1\_union::theByte.

**9.147.3.11 Q4\_State()**

```
bool azatrax::SR4::Q4_State ( ) const [inline]
```

Q4 state, return true if Q4 is closed.

References azatrax::SR4::status1\_union::Q4\_state, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateData↔Packet::status1, azatrax::SR4::status1\_union::theBits, and azatrax::SR4::status1\_union::theByte.

**9.147.3.12 RelaysOff()**

```
ErrorCode azatrax::SR4::RelaysOff (
    bool Q1,
    bool Q2,
    bool Q3,
    bool Q4 ) [inline]
```

Set output relay contacts off.

**Parameters**

<i>Q1</i>	Turn off Q1.
<i>Q2</i>	Turn off Q2.
<i>Q3</i>	Turn off Q3.
<i>Q4</i>	Turn off Q4.



References `azatrax::Azatrax::cmd_OutputRelayOff`, and `azatrax::Azatrax::send2Bytes()`.

### 9.147.3.13 RelaysOn()

```
ErrorCode azatrax::SR4::RelaysOn (
    bool Q1,
    bool Q2,
    bool Q3,
    bool Q4 ) [inline]
```

Set output relay contacts on.

#### Parameters

<i>Q1</i>	Turn on Q1.
<i>Q2</i>	Turn on Q2.
<i>Q3</i>	Turn on Q3.
<i>Q4</i>	Turn on Q4.

References `azatrax::Azatrax::cmd_OutputRelayOn`, and `azatrax::Azatrax::send2Bytes()`.

### 9.147.3.14 Sense\_1\_Latch()

```
bool azatrax::SR4::Sense_1_Latch ( ) const [inline]
```

Sense 1, return true if input line 1 was activated since last get status.

References `azatrax::SR4::status2_union::sense_1`, `azatrax::Azatrax::stateDataPacket`, `azatrax::Azatrax::StateDataPacket::status2`, `azatrax::SR4::status2_union::theBits`, and `azatrax::SR4::status2_union::theByte`.

### 9.147.3.15 Sense\_1\_Live()

```
bool azatrax::SR4::Sense_1_Live ( ) const [inline]
```

Sense 1, return true if input line 1 is now activated.

References `azatrax::SR4::status2_union::sense_1`, `azatrax::Azatrax::stateDataPacket`, `azatrax::Azatrax::StateDataPacket::status4`, `azatrax::SR4::status2_union::theBits`, and `azatrax::SR4::status2_union::theByte`.

#### 9.147.3.16 Sense\_2\_Latch()

```
bool azatrax::SR4::Sense_2_Latch ( ) const [inline]
```

Sense 2, return true if input line 2 was activated since last get status.

References azatrax::SR4::status2\_union::sense\_2, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateData←Packet::status2, azatrax::SR4::status2\_union::theBits, and azatrax::SR4::status2\_union::theByte.

#### 9.147.3.17 Sense\_2\_Live()

```
bool azatrax::SR4::Sense_2_Live ( ) const [inline]
```

Sense 2, return true if input line 2 is now activated.

References azatrax::SR4::status2\_union::sense\_2, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateData←Packet::status4, azatrax::SR4::status2\_union::theBits, and azatrax::SR4::status2\_union::theByte.

#### 9.147.3.18 Sense\_3\_Latch()

```
bool azatrax::SR4::Sense_3_Latch ( ) const [inline]
```

Sense 3, return true if input line 3 was activated since last get status.

References azatrax::SR4::status2\_union::sense\_3, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateData←Packet::status2, azatrax::SR4::status2\_union::theBits, and azatrax::SR4::status2\_union::theByte.

#### 9.147.3.19 Sense\_3\_Live()

```
bool azatrax::SR4::Sense_3_Live ( ) const [inline]
```

Sense 3, return true if input line 3 is now activated.

References azatrax::SR4::status2\_union::sense\_3, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateData←Packet::status4, azatrax::SR4::status2\_union::theBits, and azatrax::SR4::status2\_union::theByte.

### 9.147.3.20 Sense\_4\_Latch()

```
bool azatrax::SR4::Sense_4_Latch ( ) const [inline]
```

Sense 4, return true if input line 4 was activated since last get status.

References azatrax::SR4::status2\_union::sense\_4, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateDataPacket::status2, azatrax::SR4::status2\_union::theBits, and azatrax::SR4::status2\_union::theByte.

### 9.147.3.21 Sense\_4\_Live()

```
bool azatrax::SR4::Sense_4_Live ( ) const [inline]
```

Sense 4, return true if input line 4 is now activated.

References azatrax::SR4::status2\_union::sense\_4, azatrax::Azatrax::stateDataPacket, azatrax::Azatrax::StateDataPacket::status4, azatrax::SR4::status2\_union::theBits, and azatrax::SR4::status2\_union::theByte.

## 9.147.4 Friends And Related Function Documentation

### 9.147.4.1 Azatrax

```
friend class Azatrax [friend]
```

## 9.148 azatrax::Azatrax::StateDataPacket Struct Reference

Raw USB Data Packet.

```
#include <Azatrax.h>
```

## Public Attributes

- uint8\_t [commandEcho](#)  
*Command Echo byte.*
- uint8\_t [packetCount](#)  
*Packet counter (All Devices)*
- uint8\_t [status1](#)  
*Status byte 1 (All Devices)*
- uint8\_t [status2](#)  
*Status byte 2 (All Devices)*
- uint8\_t [status3](#)  
*Status byte 3 (SR4-U, SL2-U) / Stopwatch 1/100ths of a second (MRD2 only)*
- uint8\_t [status4](#)  
*Status byte 4 (SR4-U, SL2-U) / Stopwatch seconds (MRD2 only)*
- uint8\_t [stopwatchMinutes](#)  
*Stopwatch minutes (MRD2 only)*
- uint8\_t [stopwatchHours](#)  
*Stopwatch hours (MRD2 only)*
- uint8\_t [operatingMode](#)  
*Operating mode (MRD2 only)*
- uint8\_t [reserved](#) [6]  
*Reserved bytes.*
- uint8\_t [endOfData](#)  
*End of data.*

### 9.148.1 Detailed Description

Raw USB Data Packet.

This is the USB Data Packet returned in response to the GetStateData command.

### 9.148.2 Member Data Documentation

#### 9.148.2.1 [commandEcho](#)

```
uint8_t azatrax::Azatrax::StateDataPacket::commandEcho
```

Command Echo byte.

(All Devices)

### 9.148.2.2 endOfData

```
uint8_t azatrax::Azatrax::StateDataPacket::endOfData
```

End of data.

### 9.148.2.3 operatingMode

```
uint8_t azatrax::Azatrax::StateDataPacket::operatingMode
```

Operating mode (MRD2 only)

Referenced by `azatrax::MRD::OperatingMode()`.

### 9.148.2.4 packetCount

```
uint8_t azatrax::Azatrax::StateDataPacket::packetCount
```

Packet counter (All Devices)

Referenced by `azatrax::Azatrax::PacketCount()`.

### 9.148.2.5 reserved

```
uint8_t azatrax::Azatrax::StateDataPacket::reserved[6]
```

Reserved bytes.

### 9.148.2.6 status1

```
uint8_t azatrax::Azatrax::StateDataPacket::status1
```

Status byte 1 (All Devices)

See also

[MRD](#), [SL2](#), and [SR4](#) for detailed bitfields of this byte.

Referenced by `azatrax::MRD::HasRelays()`, `azatrax::MRD::Latch_1()`, `azatrax::MRD::Latch_2()`, `azatrax::SL2::Motor_1_Direction()`, `azatrax::SL2::Motor_1_State()`, `azatrax::SL2::Motor_2_Direction()`, `azatrax::SL2::Motor_2_State()`, `azatrax::SR4::Q1_State()`, `azatrax::SR4::Q2_State()`, `azatrax::SR4::Q3_State()`, `azatrax::SR4::Q4_State()`, `azatrax::MRD::Sense_1()`, and `azatrax::MRD::Sense_2()`.

### 9.148.2.7 status2

```
uint8_t azatrax::Azatrax::StateDataPacket::status2
```

Status byte 2 (All Devices)

See also

[MRD](#), [SL2](#), and [SR4](#) for detailed bitfields of this byte.

Referenced by `azatrax::MRD::AllowingExternalChanges()`, `azatrax::MRD::ExternallyChanged()`, `azatrax::MRD::Reset↵  
Status()`, `azatrax::SL2::Sense_1()`, `azatrax::SR4::Sense_1_Latch()`, `azatrax::SL2::Sense_2()`, `azatrax::SR4::Sense_2↵  
_Latch()`, `azatrax::SL2::Sense_3()`, `azatrax::SR4::Sense_3_Latch()`, `azatrax::SL2::Sense_4()`, `azatrax::SR4::Sense_↵  
4_Latch()`, and `azatrax::MRD::StopwatchTicking()`.

### 9.148.2.8 status3

```
uint8_t azatrax::Azatrax::StateDataPacket::status3
```

Status byte 3 (SR4-U, SL2-U) / Stopwatch 1/100ths of a second (MRD2 only)

Referenced by `azatrax::SL2::Input_1_Enabled()`, `azatrax::SR4::Input_1_Enabled()`, `azatrax::SL2::Input_2_Enabled()`, `azatrax::SR4::Input_2_Enabled()`, `azatrax::SL2::Input_3_Enabled()`, `azatrax::SR4::Input_3_Enabled()`, `azatrax::SL2::↵  
Input_4_Enabled()`, and `azatrax::SR4::Input_4_Enabled()`.

### 9.148.2.9 status4

```
uint8_t azatrax::Azatrax::StateDataPacket::status4
```

Status byte 4 (SR4-U, SL2-U) / Stopwatch seconds (MRD2 only)

Referenced by `azatrax::SR4::Sense_1_Live()`, `azatrax::SR4::Sense_2_Live()`, `azatrax::SR4::Sense_3_Live()`, and `azatrax::SR4::Sense_4_Live()`.

### 9.148.2.10 stopwatchHours

```
uint8_t azatrax::Azatrax::StateDataPacket::stopwatchHours
```

Stopwatch hours (MRD2 only)

Referenced by `azatrax::MRD::Stopwatch()`.

## 9.148.2.11 stopwatchMinutes

```
uint8_t azatrax::Azatrax::StateDataPacket::stopwatchMinutes
```

Stopwatch minutes (MRD2 only)

Referenced by azatrax::MRD::Stopwatch().

## 9.149 FCFSupport::Station Class Reference

The [Station](#) class implements a single station.

```
#include <Station.h>
```

### Public Member Functions

- [Station](#) ()  
*Default constructor.*
- [Station](#) (const [Station](#) &other)  
*Copy constructor, copy from another station instance.*
- [Station](#) & operator= ([Station](#) &other)  
*Assignment operator, copy from another station instance.*
- [Station](#) (const char \*n, [Division](#) \*d, const char \*c)  
*Full constructor.*
- ~[Station](#) ()  
*Destructor.*
- const char \* [Name](#) () const  
*Return the station's name.*
- [Division](#) \* [MyDivision](#) () const  
*Return the station's division.*
- const char \* [Comment](#) () const  
*Return the station's comment.*
- int [NumberOfIndustries](#) () const  
*Return the number of industries at this station.*
- [FCFSupport::Industry](#) \* [TheIndustry](#) (int i) const  
*Return the lth industry at this station.*
- int [AppendIndustry](#) ([FCFSupport::Industry](#) \*industry)  
*Append an industry to this station's list of industries.*

### Private Attributes

- string [name](#)  
*The station's name.*
- string [comment](#)  
*The station's comment.*
- [Division](#) \* [division](#)  
*The station's division.*
- vector< [Industry](#) \* > [industries](#)  
*The list of industries at this station.*

## Friends

- class [System](#)

*The [System](#) class is a friend.*

### 9.149.1 Detailed Description

The [Station](#) class implements a single station.

A station exists within a division and contains a number of industries. A [Station](#) has a name, a comment, it belongs to a division, and has a list of industries.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.149.2 Constructor & Destructor Documentation

#### 9.149.2.1 [Station\(\)](#) [1/3]

```
FCFSupport::Station::Station ( ) [inline]
```

Default constructor.

Initialize all slots to empty values.

References comment, division, and name.

#### 9.149.2.2 [Station\(\)](#) [2/3]

```
FCFSupport::Station::Station (
    const Station & other ) [inline]
```

Copy constructor, copy from another station instance.

#### Parameters

<i>other</i>	The other station instance.
--------------	-----------------------------

References comment, division, industries, and name.



### 9.149.2.3 Station() [3/3]

```
FCFSupport::Station::Station (
    const char * n,
    Division * d,
    const char * c ) [inline]
```

Full constructor.

Create a fresh station instance, given a name, division, and a comment. Initially, the industry list is empty.

#### Parameters

<i>n</i>	The new station's name.
<i>d</i>	The division the station belongs to.
<i>c</i>	A comment string.

References comment, division, and name.

### 9.149.2.4 ~Station()

```
FCFSupport::Station::~~Station ( ) [inline]
```

Destructor.

## 9.149.3 Member Function Documentation

### 9.149.3.1 AppendIndustry()

```
int FCFSupport::Station::AppendIndustry (
    FCFSupport::Industry * industry ) [inline]
```

Append an industry to this station's list of industries.

#### Parameters

<i>industry</i>	The industry to append.
-----------------	-------------------------

References industries.

#### 9.149.3.2 Comment()

```
const char* FCFSupport::Station::Comment ( ) const [inline]
```

Return the station's comment.

References comment.

#### 9.149.3.3 MyDivision()

```
Division* FCFSupport::Station::MyDivision ( ) const [inline]
```

Return the station's division.

References division.

#### 9.149.3.4 Name()

```
const char* FCFSupport::Station::Name ( ) const [inline]
```

Return the station's name.

References name.

#### 9.149.3.5 NumberOfIndustries()

```
int FCFSupport::Station::NumberOfIndustries ( ) const [inline]
```

Return the number of industries at this station.

References industries.

#### 9.149.3.6 operator=()

```
Station& FCFSupport::Station::operator= (   
    Station & other ) [inline]
```

Assignment operator, copy from another station instance.

## Parameters

<i>other</i>	The other station instance.
--------------	-----------------------------

References comment, division, industries, and name.

## 9.149.3.7 TheIndustry()

```
FCFSupport::Industry* FCFSupport::Station::TheIndustry (  
    int i ) const [inline]
```

Return the lth industry at this station.

## Parameters

<i>i</i>	The industry index.
----------	---------------------

References i, and industries.

## 9.149.4 Friends And Related Function Documentation

## 9.149.4.1 System

```
friend class System [friend]
```

The [System](#) class is a friend.

## 9.149.5 Member Data Documentation

## 9.149.5.1 comment

```
string FCFSupport::Station::comment [private]
```

The station's comment.

Referenced by [Comment\(\)](#), [operator=\(\)](#), and [Station\(\)](#).

#### 9.149.5.2 division

```
Division* FCFSupport::Station::division [private]
```

The station's division.

Referenced by `MyDivision()`, `operator=()`, and `Station()`.

#### 9.149.5.3 industries

```
vector<Industry *> FCFSupport::Station::industries [private]
```

The list of industries at this station.

Referenced by `AppendIndustry()`, `NumberOfIndustries()`, `operator=()`, `Station()`, and `TheIndustry()`.

#### 9.149.5.4 name

```
string FCFSupport::Station::name [private]
```

The station's name.

Referenced by `Name()`, `operator=()`, and `Station()`.

### 9.150 TTSupport::Station Class Reference

The `Station` class implements a station.

```
#include <Station.h>
```

## Public Member Functions

- [Station](#) (string name\_="Unknown", double smile\_=0)  
*Construct a station object, given a name and a scale mile location.*
- [Station](#) (const [Station](#) &other)  
*Copy constructor.*
- [Station](#) & [operator=](#) (const [Station](#) &other)  
*Assignment operator.*
- [~Station](#) ()  
*Destructor.*
- const char \* [Name](#) () const  
*Return the name of the station.*
- double [SMile](#) () const  
*Return the scale mile of the station.*
- int [DuplicateStationIndex](#) () const  
*Return the duplicate station index.*
- void [SetDuplicateStationIndex](#) (int index)  
*Set the duplication station index.*
- [StorageTrack](#) \* [AddStorageTrack](#) (string name\_)  
*Add a storage track.*
- [StorageTrack](#) \* [FindStorageTrack](#) (string name)  
*Find a storage track by name.*
- [StorageTrack](#) \* [FindTrackTrainsStoredOn](#) (string trainNumber, double fromtime, double totime)  
*Find track a train is stored on.*
- int [NumberOfStorageTracks](#) () const  
*Number of storage tracks.*
- [StorageTrackMap::const\\_iterator](#) [FirstStorageTrack](#) () const  
*Return a const\_iterator for the first element in the storage track map.*
- [StorageTrackMap::const\\_iterator](#) [LastStorageTrack](#) () const  
*Return a const\_iterator for the last element in the storage track map.*
- ostream & [Write](#) (ostream &stream) const  
*Write object to a stream.*
- istream & [Read](#) (istream &stream)  
*Read an object from a stream.*

## Private Attributes

- string [name](#)  
*Station name.*
- [StorageTrackMap](#) [storageTracks](#)  
*Storage track map.*
- double [smile](#)  
*Scale Mile.*
- int [duplicateStationIndex](#)  
*Duplicate station index.*

### 9.150.1 Detailed Description

The [Station](#) class implements a station.

Stations are not specifically passenger stations, but are any place where trains stop or meet or might just be important mile post locations used for time keeping checks. They also can be just sidings.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.150.2 Constructor & Destructor Documentation

#### 9.150.2.1 [Station\(\)](#) [1/2]

```
TTSupport::Station::Station (
    string name_ = "Unknown",
    double smile_ = 0 ) [inline]
```

Construct a station object, given a name and a scale mile location.

#### Parameters

<i>name</i> ↔ —	The name of the station.
<i>smile</i> ↔ —	The scale mile location of the station.

#### 9.150.2.2 [Station\(\)](#) [2/2]

```
TTSupport::Station::Station (
    const Station & other ) [inline]
```

Copy constructor.

Copy one station to another.

#### Parameters

<i>other</i>	The other station.
--------------	--------------------

References `duplicateStationIndex`, `name`, `smile`, and `storageTracks`.

### 9.150.2.3 ~Station()

```
TTSupport::Station::~~Station ( ) [inline]
```

Destructor.

## 9.150.3 Member Function Documentation

### 9.150.3.1 AddStorageTrack()

```
StorageTrack* TTSupport::Station::AddStorageTrack (
    string name_ )
```

Add a storage track.

#### Parameters

<i>name</i> ↔	The name of the storage track.
—	

### 9.150.3.2 DuplicateStationIndex()

```
int TTSupport::Station::DuplicateStationIndex ( ) const [inline]
```

Return the duplicate station index.

This is the index of another station that is the physical duplicate of this one. Only meaningful on out-and-back type layouts or other layout configurations where stations are logically duplicated due to trackage having dual meaning.

### 9.150.3.3 FindStorageTrack()

```
StorageTrack* TTSupport::Station::FindStorageTrack (
    string name ) [inline]
```

Find a storage track by name.

**Parameters**

<i>name</i>	The name of the storage track.
-------------	--------------------------------

**9.150.3.4 FindTrackTrainsStoredOn()**

```
StorageTrack* TTSupport::Station::FindTrackTrainIsStoredOn (
    string trainNumber,
    double fromtime,
    double totime )
```

Find track a train is stored on.

**Parameters**

<i>trainNumber</i>	The train number (symbol) to search for.
<i>fromtime</i>	The from time to check.
<i>totime</i>	The to time to check.

**9.150.3.5 FirstStorageTrack()**

```
StorageTrackMap::const_iterator TTSupport::Station::FirstStorageTrack ( ) const [inline]
```

Return a `const_iterator` for the first element in the storage track map.

**9.150.3.6 LastStorageTrack()**

```
StorageTrackMap::const_iterator TTSupport::Station::LastStorageTrack ( ) const [inline]
```

Return a `const_iterator` for the last element in the storage track map.

References `TTSupport::Occupied::Read()`, `lcc::stream`, and `TTSupport::Occupied::Write()`.

**9.150.3.7 Name()**

```
const char* TTSupport::Station::Name ( ) const [inline]
```

Return the name of the station.



### 9.150.3.8 NumberOfStorageTracks()

```
int TTSupport::Station::NumberOfStorageTracks ( ) const [inline]
```

Number of storage tracks.

### 9.150.3.9 operator=()

```
Station& TTSupport::Station::operator= (
    const Station & other ) [inline]
```

Assignment operator.

Assign one station to another.

#### Parameters

<i>other</i>	The other station.
--------------	--------------------

References duplicateStationIndex, name, smile, and storageTracks.

### 9.150.3.10 Read()

```
istream& TTSupport::Station::Read (
    istream & stream )
```

Read an object from a stream.

#### Parameters

<i>stream</i>	Stream to read from.
---------------	----------------------

### 9.150.3.11 SetDuplicateStationIndex()

```
void TTSupport::Station::SetDuplicateStationIndex (
    int index ) [inline]
```

Set the duplication station index.

**Parameters**

<i>index</i>	The index of the duplicate station.
--------------	-------------------------------------

**9.150.3.12 SMile()**

```
double TTSupport::Station::SMile ( ) const [inline]
```

Return the scale mile of the station.

**9.150.3.13 Write()**

```
ostream& TTSupport::Station::Write (
    ostream & stream ) const
```

Write object to a stream.

**Parameters**

<i>stream</i>	Stream to write to.
---------------	---------------------

**9.150.4 Member Data Documentation****9.150.4.1 duplicateStationIndex**

```
int TTSupport::Station::duplicateStationIndex [private]
```

Duplicate station index.

Referenced by operator=(), and Station().

**9.150.4.2 name**

```
string TTSupport::Station::name [private]
```

[Station](#) name.

Referenced by operator=(), and Station().

#### 9.150.4.3 smile

```
double TTSupport::Station::smile [private]
```

Scale Mile.

Referenced by operator=(), and Station().

#### 9.150.4.4 storageTracks

```
StorageTrackMap TTSupport::Station::storageTracks [private]
```

Storage track map.

Referenced by operator=(), and Station().

## 9.151 FCFSupport::SwitchListElement::StationOrIndustry Union Reference

A const pointer to a train's stop, which can be either a station or an industry, depending on what kind of train it is.

```
#include <SwitchList.h>
```

### Public Attributes

- const [Station](#) \* [station](#)  
*A station stop, for Box Moves and Way Freights.*
- const [Industry](#) \* [industry](#)  
*An industry stop, for Manifest Freights.*

### 9.151.1 Detailed Description

A const pointer to a train's stop, which can be either a station or an industry, depending on what kind of train it is.

### 9.151.2 Member Data Documentation

### 9.151.2.1 industry

```
const Industry* FCFSupport::SwitchListElement::StationOrIndustry::industry
```

An industry stop, for Manifest Freights.

Referenced by FCFSupport::SwitchListElement::DropStopEQ(), FCFSupport::SwitchListElement::DropStopIndustry(), FCFSupport::SwitchListElement::DropStopStation(), FCFSupport::SwitchListElement::operator=(), and FCFSupport::SwitchListElement::SwitchListElement().

### 9.151.2.2 station

```
const Station* FCFSupport::SwitchListElement::StationOrIndustry::station
```

A station stop, for Box Moves and Way Freights.

Referenced by FCFSupport::SwitchListElement::DropStopEQ(), FCFSupport::SwitchListElement::DropStopStation(), and FCFSupport::SwitchListElement::SwitchListElement().

## 9.152 FCFSupport::Train::StationOrIndustry Union Reference

Union of stations or industries, used for stops.

```
#include <Train.h>
```

### Public Attributes

- [Station](#) \* [station](#)  
*[Station](#), for other than Manifest freights.*
- [Industry](#) \* [industry](#)  
*[Industry](#), for Manifest freights.*

### 9.152.1 Detailed Description

Union of stations or industries, used for stops.

### 9.152.2 Member Data Documentation

## 9.152.2.1 industry

```
Industry* FCFSupport::Train::StationOrIndustry::industry
```

[Industry](#), for Manifest freights.

## 9.152.2.2 station

```
Station* FCFSupport::Train::StationOrIndustry::station
```

[Station](#), for other then Manifest freights.

## 9.153 TTSupport::StationTimes Class Reference

[Station](#) times class, used by the LaTeX generator methods.

```
#include <TimeTableSystem.h>
```

## Public Member Functions

- [StationTimes](#) (double a=-1, double d=-1, [Stop::FlagType](#) f=[Stop::Transit](#))  
*Constructor: create an entry for a time table cell.*
- [StationTimes](#) (const [StationTimes](#) &other)  
*Copy constructor: create an entry for a time table cell from an existing [StationTimes](#) object.*
- [StationTimes](#) & operator= (const [StationTimes](#) &other)  
*Assignment operator: copy the fields from another [StationTimes](#) object.*
- double [Arrival](#) () const  
*Accessor for the arrival time.*
- double [Departure](#) () const  
*Accessor for the departure time.*
- [Stop::FlagType](#) [Flag](#) () const  
*Accessor for the type of stop flag.*

## Private Attributes

- double [arrival](#)  
*The arrival time, in scale time units.*
- double [departure](#)  
*The departure time, in scale time units.*
- [Stop::FlagType](#) [flag](#)  
*The stop flag: Origin, Terminate, or Transit.*

### 9.153.1 Detailed Description

[Station](#) times class, used by the LaTeX generator methods.

This class holds time table information used in the code that generates the LaTeX tables. Each [StationTimes](#) item contains one table element in the form of an arrival time and a departure time. The flag member indicates if only the arrival time, departure time, or both times are valid. An originating train has no arrival time and a terminating train has no departure time.

This class is actually used to hold the information for a single cell in a formatted time table. Each cell contains an arrival time and a departure time. Each row in the table contains the information for a specific station and each column contains the information for a single train.

See also

[TrainStationTimes](#) [TrainTimesAtStation](#).

Author

Robert Heller <heller@deepsoft.com>

### 9.153.2 Constructor & Destructor Documentation

#### 9.153.2.1 [StationTimes\(\)](#) [1/2]

```
TTSupport::StationTimes::StationTimes (
    double a = -1,
    double d = -1,
    Stop::FlagType f = Stop::Transit ) [inline]
```

Constructor: create an entry for a time table cell.

Parameters

<i>a</i>	The arrival time.
<i>d</i>	The departure time.
<i>f</i>	The stop flag: Origin, Terminate, or Transit.

#### 9.153.2.2 [StationTimes\(\)](#) [2/2]

```
TTSupport::StationTimes::StationTimes (
    const StationTimes & other ) [inline]
```

Copy constructor: create an entry for a time table cell from an existing [StationTimes](#) object.

#### Parameters

<i>other</i>	The other <a href="#">StationTimes</a> object.
--------------	--

References arrival, departure, and flag.

### 9.153.3 Member Function Documentation

#### 9.153.3.1 Arrival()

```
double TTSupport::StationTimes::Arrival ( ) const [inline]
```

Accessor for the arrival time.

#### 9.153.3.2 Departure()

```
double TTSupport::StationTimes::Departure ( ) const [inline]
```

Accessor for the departure time.

#### 9.153.3.3 Flag()

```
Stop::FlagType TTSupport::StationTimes::Flag ( ) const [inline]
```

Accessor for the type of stop flag.

#### 9.153.3.4 operator=()

```
StationTimes& TTSupport::StationTimes::operator= (  
    const StationTimes & other ) [inline]
```

Assignment operator: copy the fields from another [StationTimes](#) object.

## Parameters

<i>other</i>	The other <a href="#">StationTimes</a> object.
--------------	--

References arrival, departure, and flag.

## 9.153.4 Member Data Documentation

### 9.153.4.1 arrival

```
double TTSupport::StationTimes::arrival [private]
```

The arrival time, in scale time units.

Only used for trains in transit and for terminating trains.

Referenced by operator=(), and StationTimes().

### 9.153.4.2 departure

```
double TTSupport::StationTimes::departure [private]
```

The departure time, in scale time units.

Only used for trains in transit and for originating trains.

Referenced by operator=(), and StationTimes().

### 9.153.4.3 flag

```
Stop::FlagType TTSupport::StationTimes::flag [private]
```

The stop flag: Origin, Terminate, or Transit.

Referenced by operator=(), and StationTimes().

## 9.154 azatrax::SR4::status1\_union Union Reference

Status byte 1 union type (Outputs)



## Public Attributes

- uint8\_t [theByte](#)  
*Status byte as a byte.*
- struct {  
    unsigned int [Q1\\_state](#):1  
        *Q1 state.*  
    unsigned int [Q2\\_state](#):1  
        *Q2 state.*  
    unsigned int [Q3\\_state](#):1  
        *Q3 state.*  
    unsigned int [Q4\\_state](#):1  
        *Q4 state.*  
    unsigned int [reserved](#):4  
        *reserved bits*  
} [theBits](#)

*Status byte as bit fields.*

### 9.154.1 Detailed Description

Status byte 1 union type (Outputs)

### 9.154.2 Member Data Documentation

#### 9.154.2.1 Q1\_state

```
unsigned int azatrax::SR4::status1_union::Q1_state
```

Q1 state.

Referenced by azatrax::SR4::Q1\_State().

#### 9.154.2.2 Q2\_state

```
unsigned int azatrax::SR4::status1_union::Q2_state
```

Q2 state.

Referenced by azatrax::SR4::Q2\_State().

#### 9.154.2.3 Q3\_state

```
unsigned int azatrax::SR4::status1_union::Q3_state
```

Q3 state.

Referenced by azatrax::SR4::Q3\_State().

#### 9.154.2.4 Q4\_state

```
unsigned int azatrax::SR4::status1_union::Q4_state
```

Q4 state.

Referenced by azatrax::SR4::Q4\_State().

#### 9.154.2.5 reserved

```
unsigned int azatrax::SR4::status1_union::reserved
```

reserved bits

#### 9.154.2.6 theBits

```
struct { ... } azatrax::SR4::status1_union::theBits
```

Status byte as bit fields.

Referenced by azatrax::SR4::Q1\_State(), azatrax::SR4::Q2\_State(), azatrax::SR4::Q3\_State(), and azatrax::SR4::Q4\_State().

#### 9.154.2.7 theByte

```
uint8_t azatrax::SR4::status1_union::theByte
```

Status byte as a byte.

Referenced by azatrax::SR4::Q1\_State(), azatrax::SR4::Q2\_State(), azatrax::SR4::Q3\_State(), and azatrax::SR4::Q4\_State().

## 9.155 azatrax::SL2::status1\_union Union Reference

Status byte 1 union type (Output states)

### Public Attributes

- `uint8_t theByte`  
*Status byte as a byte.*
- `struct {`
  - `unsigned int motor_1_direction:1`  
*Motor 1 direction.*
  - `unsigned int motor_1_state:1`  
*Motor 1 state.*
  - `unsigned int motor_2_direction:1`  
*Motor 2 direction.*
  - `unsigned int motor_2_state:1`  
*Motor 2 state.*
  - `unsigned int reserved:4`  
*Reserved bits.*`} theBits`

*Status byte as bit fields.*

### 9.155.1 Detailed Description

Status byte 1 union type (Output states)

### 9.155.2 Member Data Documentation

#### 9.155.2.1 motor\_1\_direction

```
unsigned int azatrax::SL2::status1_union::motor_1_direction
```

Motor 1 direction.

Referenced by `azatrax::SL2::Motor_1_Direction()`.

#### 9.155.2.2 motor\_1\_state

```
unsigned int azatrax::SL2::status1_union::motor_1_state
```

Motor 1 state.

Referenced by azatrax::SL2::Motor\_1\_State().

#### 9.155.2.3 motor\_2\_direction

```
unsigned int azatrax::SL2::status1_union::motor_2_direction
```

Motor 2 direction.

Referenced by azatrax::SL2::Motor\_2\_Direction().

#### 9.155.2.4 motor\_2\_state

```
unsigned int azatrax::SL2::status1_union::motor_2_state
```

Motor 2 state.

Referenced by azatrax::SL2::Motor\_2\_State().

#### 9.155.2.5 reserved

```
unsigned int azatrax::SL2::status1_union::reserved
```

Reserved bits.

#### 9.155.2.6 theBits

```
struct { ... } azatrax::SL2::status1_union::theBits
```

Status byte as bit fields.

Referenced by azatrax::SL2::Motor\_1\_Direction(), azatrax::SL2::Motor\_1\_State(), azatrax::SL2::Motor\_2\_Direction(), and azatrax::SL2::Motor\_2\_State().

### 9.155.2.7 theByte

```
uint8_t azatrax::SL2::status1_union::theByte
```

Status byte as a byte.

Referenced by azatrax::SL2::Motor\_1\_Direction(), azatrax::SL2::Motor\_1\_State(), azatrax::SL2::Motor\_2\_Direction(), and azatrax::SL2::Motor\_2\_State().

## 9.156 azatrax::MRD::status1\_union Union Reference

Status byte 1 union type.

### Public Attributes

- `uint8_t theByte`  
*Status byte as a byte.*
  - struct {
    - unsigned int `sense_1`:1  
*Sense 1.*
    - unsigned int `sense_2`:1  
*Sense 2.*
    - unsigned int `latch_1`:1  
*Latch 1.*
    - unsigned int `latch_2`:1  
*Latch 2.*
    - unsigned int `modtype`:1  
*Module type.*
    - unsigned int `reserved`:3  
*Reserved bits.*
- } `theBits`

*Status byte as bit fields.*

### 9.156.1 Detailed Description

Status byte 1 union type.

### 9.156.2 Member Data Documentation

**9.156.2.1 latch\_1**

```
unsigned int azatrax::MRD::status1_union::latch_1
```

Latch 1.

Referenced by azatrax::MRD::Latch\_1().

**9.156.2.2 latch\_2**

```
unsigned int azatrax::MRD::status1_union::latch_2
```

Latch 2.

Referenced by azatrax::MRD::Latch\_2().

**9.156.2.3 modtype**

```
unsigned int azatrax::MRD::status1_union::modtype
```

Module type.

Referenced by azatrax::MRD::HasRelays().

**9.156.2.4 reserved**

```
unsigned int azatrax::MRD::status1_union::reserved
```

Reserved bits.

**9.156.2.5 sense\_1**

```
unsigned int azatrax::MRD::status1_union::sense_1
```

Sense 1.

Referenced by azatrax::MRD::Sense\_1().

## 9.156.2.6 sense\_2

```
unsigned int azatrax::MRD::status1_union::sense_2
```

Sense 2.

Referenced by azatrax::MRD::Sense\_2().

## 9.156.2.7 theBits

```
struct { ... } azatrax::MRD::status1_union::theBits
```

Status byte as bit fields.

Referenced by azatrax::MRD::HasRelays(), azatrax::MRD::Latch\_1(), azatrax::MRD::Latch\_2(), azatrax::MRD::Sense\_1(), and azatrax::MRD::Sense\_2().

## 9.156.2.8 theByte

```
uint8_t azatrax::MRD::status1_union::theByte
```

Status byte as a byte.

Referenced by azatrax::MRD::HasRelays(), azatrax::MRD::Latch\_1(), azatrax::MRD::Latch\_2(), azatrax::MRD::Sense\_1(), and azatrax::MRD::Sense\_2().

## 9.157 azatrax::MRD::status2\_union Union Reference

Status byte 2 union type.

## Public Attributes

- uint8\_t [theByte](#)  
*Status byte as a byte.*
- struct {
  - unsigned int [resetStatus](#):1  
*Reset Status.*
  - unsigned int [stopwatchTicking](#):1  
*Stopwatch Ticking.*
  - unsigned int [externallyChanged](#):1  
*Externally Changed.*
  - unsigned int [allowExternalChanges](#):1  
*Allow External Changes.*
  - unsigned int [reserved](#):4  
*Reserved bits.*

} [theBits](#)

*Status byte as bit fields.*

### 9.157.1 Detailed Description

Status byte 2 union type.

### 9.157.2 Member Data Documentation

#### 9.157.2.1 allowExternalChanges

```
unsigned int azatrax::MRD::status2_union::allowExternalChanges
```

Allow External Changes.

Referenced by azatrax::MRD::AllowingExternalChanges().

#### 9.157.2.2 externallyChanged

```
unsigned int azatrax::MRD::status2_union::externallyChanged
```

Externally Changed.

Referenced by azatrax::MRD::ExternallyChanged().

#### 9.157.2.3 reserved

```
unsigned int azatrax::MRD::status2_union::reserved
```

Reserved bits.

#### 9.157.2.4 resetStatus

```
unsigned int azatrax::MRD::status2_union::resetStatus
```

Reset Status.

Referenced by azatrax::MRD::ResetStatus().



## 9.157.2.5 stopwatchTicking

```
unsigned int azatrax::MRD::status2_union::stopwatchTicking
```

Stopwatch Ticking.

Referenced by azatrax::MRD::StopwatchTicking().

## 9.157.2.6 theBits

```
struct { ... } azatrax::MRD::status2_union::theBits
```

Status byte as bit fields.

Referenced by azatrax::MRD::AllowingExternalChanges(), azatrax::MRD::ExternallyChanged(), azatrax::MRD::Reset↵  
Status(), and azatrax::MRD::StopwatchTicking().

## 9.157.2.7 theByte

```
uint8_t azatrax::MRD::status2_union::theByte
```

Status byte as a byte.

Referenced by azatrax::MRD::AllowingExternalChanges(), azatrax::MRD::ExternallyChanged(), azatrax::MRD::Reset↵  
Status(), and azatrax::MRD::StopwatchTicking().

## 9.158 azatrax::SR4::status2\_union Union Reference

Status byte 2 union type (Input sense)

## Public Attributes

- uint8\_t [theByte](#)  
*Status byte as a byte.*
- struct {
  - unsigned int [sense\\_1](#):1  
*Sense 1.*
  - unsigned int [sense\\_2](#):1  
*Sense 2.*
  - unsigned int [sense\\_3](#):1  
*Sense 3.*
  - unsigned int [sense\\_4](#):1  
*Sense 4.*
  - unsigned int [reserved](#):4  
*Reserved bits.*

} [theBits](#)

*Status byte as bit fields.*

### 9.158.1 Detailed Description

Status byte 2 union type (Input sense)

### 9.158.2 Member Data Documentation

#### 9.158.2.1 reserved

```
unsigned int azatrax::SR4::status2_union::reserved
```

Reserved bits.

#### 9.158.2.2 sense\_1

```
unsigned int azatrax::SR4::status2_union::sense_1
```

Sense 1.

Referenced by azatrax::SR4::Sense\_1\_Latch(), and azatrax::SR4::Sense\_1\_Live().

#### 9.158.2.3 sense\_2

```
unsigned int azatrax::SR4::status2_union::sense_2
```

Sense 2.

Referenced by azatrax::SR4::Sense\_2\_Latch(), and azatrax::SR4::Sense\_2\_Live().

#### 9.158.2.4 sense\_3

```
unsigned int azatrax::SR4::status2_union::sense_3
```

Sense 3.

Referenced by azatrax::SR4::Sense\_3\_Latch(), and azatrax::SR4::Sense\_3\_Live().

## 9.158.2.5 sense\_4

```
unsigned int azatrax::SR4::status2_union::sense_4
```

Sense 4.

Referenced by azatrax::SR4::Sense\_4\_Latch(), and azatrax::SR4::Sense\_4\_Live().

## 9.158.2.6 theBits

```
struct { ... } azatrax::SR4::status2_union::theBits
```

Status byte as bit fields.

Referenced by azatrax::SR4::Sense\_1\_Latch(), azatrax::SR4::Sense\_1\_Live(), azatrax::SR4::Sense\_2\_Latch(), azatrax::SR4::Sense\_2\_Live(), azatrax::SR4::Sense\_3\_Latch(), azatrax::SR4::Sense\_3\_Live(), azatrax::SR4::Sense\_4\_Latch(), and azatrax::SR4::Sense\_4\_Live().

## 9.158.2.7 theByte

```
uint8_t azatrax::SR4::status2_union::theByte
```

Status byte as a byte.

Referenced by azatrax::SR4::Sense\_1\_Latch(), azatrax::SR4::Sense\_1\_Live(), azatrax::SR4::Sense\_2\_Latch(), azatrax::SR4::Sense\_2\_Live(), azatrax::SR4::Sense\_3\_Latch(), azatrax::SR4::Sense\_3\_Live(), azatrax::SR4::Sense\_4\_Latch(), and azatrax::SR4::Sense\_4\_Live().

## 9.159 azatrax::SL2::status2\_union Union Reference

Status byte 2 union type (Input sense)

## Public Attributes

- uint8\_t [theByte](#)  
*Status byte as a byte.*
- struct {
  - unsigned int [sense\\_1](#):1  
*Sense 1.*
  - unsigned int [sense\\_2](#):1  
*Sense 2.*
  - unsigned int [sense\\_3](#):1  
*Sense 3.*
  - unsigned int [sense\\_4](#):1  
*Sense 4.*
  - unsigned int [reserved](#):4  
*Reserved bits.*

} [theBits](#)

*Status byte as bit fields.*

### 9.159.1 Detailed Description

Status byte 2 union type (Input sense)

### 9.159.2 Member Data Documentation

#### 9.159.2.1 reserved

```
unsigned int azatrax::SL2::status2_union::reserved
```

Reserved bits.

#### 9.159.2.2 sense\_1

```
unsigned int azatrax::SL2::status2_union::sense_1
```

Sense 1.

Referenced by azatrax::SL2::Sense\_1().

#### 9.159.2.3 sense\_2

```
unsigned int azatrax::SL2::status2_union::sense_2
```

Sense 2.

Referenced by azatrax::SL2::Sense\_2().

#### 9.159.2.4 sense\_3

```
unsigned int azatrax::SL2::status2_union::sense_3
```

Sense 3.

Referenced by azatrax::SL2::Sense\_3().

## 9.159.2.5 sense\_4

```
unsigned int azatrax::SL2::status2_union::sense_4
```

Sense 4.

Referenced by azatrax::SL2::Sense\_4().

## 9.159.2.6 theBits

```
struct { ... } azatrax::SL2::status2_union::theBits
```

Status byte as bit fields.

Referenced by azatrax::SL2::Sense\_1(), azatrax::SL2::Sense\_2(), azatrax::SL2::Sense\_3(), and azatrax::SL2::Sense\_4().

## 9.159.2.7 theByte

```
uint8_t azatrax::SL2::status2_union::theByte
```

Status byte as a byte.

Referenced by azatrax::SL2::Sense\_1(), azatrax::SL2::Sense\_2(), azatrax::SL2::Sense\_3(), and azatrax::SL2::Sense\_4().

## 9.160 azatrax::SR4::status3\_union Union Reference

Status byte 3 union type (Input Control Status)

## Public Attributes

- `uint8_t theByte`  
*Status byte as a byte.*
- `struct {`
  - `unsigned int input_1_enabled:1`  
*Input 1 enabled?*
  - `unsigned int input_2_enabled:1`  
*Input 2 enabled?*
  - `unsigned int input_3_enabled:1`  
*Input 3 enabled?*
  - `unsigned int input_4_enabled:1`  
*Input 4 enabled?*
  - `unsigned int reserved:4`  
*Reserved bits.*
- `} theBits`

*Status byte as bit fields.*

### 9.160.1 Detailed Description

Status byte 3 union type (Input Control Status)

### 9.160.2 Member Data Documentation

#### 9.160.2.1 input\_1\_enabled

```
unsigned int azatrax::SR4::status3_union::input_1_enabled
```

Input 1 enabled?

Referenced by azatrax::SR4::Input\_1\_Enabled().

#### 9.160.2.2 input\_2\_enabled

```
unsigned int azatrax::SR4::status3_union::input_2_enabled
```

Input 2 enabled?

Referenced by azatrax::SR4::Input\_2\_Enabled().

#### 9.160.2.3 input\_3\_enabled

```
unsigned int azatrax::SR4::status3_union::input_3_enabled
```

Input 3 enabled?

Referenced by azatrax::SR4::Input\_3\_Enabled().

#### 9.160.2.4 input\_4\_enabled

```
unsigned int azatrax::SR4::status3_union::input_4_enabled
```

Input 4 enabled?

Referenced by azatrax::SR4::Input\_4\_Enabled().

## 9.160.2.5 reserved

```
unsigned int azatrax::SR4::status3_union::reserved
```

Reserved bits.

## 9.160.2.6 theBits

```
struct { ... } azatrax::SR4::status3_union::theBits
```

Status byte as bit fields.

Referenced by azatrax::SR4::Input\_1\_Enabled(), azatrax::SR4::Input\_2\_Enabled(), azatrax::SR4::Input\_3\_Enabled(), and azatrax::SR4::Input\_4\_Enabled().

## 9.160.2.7 theByte

```
uint8_t azatrax::SR4::status3_union::theByte
```

Status byte as a byte.

Referenced by azatrax::SR4::Input\_1\_Enabled(), azatrax::SR4::Input\_2\_Enabled(), azatrax::SR4::Input\_3\_Enabled(), and azatrax::SR4::Input\_4\_Enabled().

## 9.161 azatrax::SL2::status3\_union Union Reference

Status byte 3 union type (Input control state)

## Public Attributes

- `uint8_t theByte`  
*Status byte as a byte.*
- `struct {`
  - `unsigned int input_1_enabled:1`  
*Input 1 enabled?*
  - `unsigned int input_2_enabled:1`  
*Input 2 enabled?*
  - `unsigned int input_3_enabled:1`  
*Input 3 enabled?*
  - `unsigned int input_4_enabled:1`  
*Input 4 enabled?*
  - `unsigned int reserved:4`  
*Reserved bits.*
- `} theBits`

*Status byte as bit fields.*

### 9.161.1 Detailed Description

Status byte 3 union type (Input control state)

### 9.161.2 Member Data Documentation

#### 9.161.2.1 input\_1\_enabled

```
unsigned int azatrax::SL2::status3_union::input_1_enabled
```

Input 1 enabled?

Referenced by azatrax::SL2::Input\_1\_Enabled().

#### 9.161.2.2 input\_2\_enabled

```
unsigned int azatrax::SL2::status3_union::input_2_enabled
```

Input 2 enabled?

Referenced by azatrax::SL2::Input\_2\_Enabled().

#### 9.161.2.3 input\_3\_enabled

```
unsigned int azatrax::SL2::status3_union::input_3_enabled
```

Input 3 enabled?

Referenced by azatrax::SL2::Input\_3\_Enabled().

#### 9.161.2.4 input\_4\_enabled

```
unsigned int azatrax::SL2::status3_union::input_4_enabled
```

Input 4 enabled?

Referenced by azatrax::SL2::Input\_4\_Enabled().



#### 9.161.2.5 reserved

```
unsigned int azatrax::SL2::status3_union::reserved
```

Reserved bits.

#### 9.161.2.6 theBits

```
struct { ... } azatrax::SL2::status3_union::theBits
```

Status byte as bit fields.

Referenced by azatrax::SL2::Input\_1\_Enabled(), azatrax::SL2::Input\_2\_Enabled(), azatrax::SL2::Input\_3\_Enabled(), and azatrax::SL2::Input\_4\_Enabled().

#### 9.161.2.7 theByte

```
uint8_t azatrax::SL2::status3_union::theByte
```

Status byte as a byte.

Referenced by azatrax::SL2::Input\_1\_Enabled(), azatrax::SL2::Input\_2\_Enabled(), azatrax::SL2::Input\_3\_Enabled(), and azatrax::SL2::Input\_4\_Enabled().

## 9.162 TTSupport::Stop Class Reference

This class implements a stop.

```
#include <Train.h>
```

### Public Types

- enum [FlagType](#) { [Origin](#), [Terminate](#), [Transit](#) }

*Type of stop.*

## Public Member Functions

- [Stop](#) (int stationindex\_=0, [FlagType](#) flag\_=Origin)  
*Constructor: create a new stop.*
- [Stop](#) (const [Stop](#) &other)  
*Copy constructor, create a stop from another stop.*
- [Stop](#) & [operator=](#) (const [Stop](#) &other)  
*Assignment operator, assign one stop to another stop.*
- [~Stop](#) ()  
*Destructor.*
- double [Layover](#) () const  
*Return layover period.*
- void [SetLayover](#) (double period)  
*Update layover period.*
- double [Departure](#) (double arrival) const  
*Return departure time.*
- int [StationIndex](#) () const  
*Return the station index.*
- [Cab](#) \* [TheCab](#) () const  
*Return the cab.*
- void [SetCab](#) ([Cab](#) \*newcab)  
*Update the cab.*
- int [NumberOfNotes](#) () const  
*Return the number of notes.*
- int [Note](#) (int i) const  
*Return the ith note.*
- void [AddNote](#) (int note)  
*Add a note.*
- void [RemoveNote](#) (int note)  
*Remove note.*
- [FlagType](#) [Flag](#) () const  
*Return the flag.*
- const char \* [StorageTrackName](#) () const  
*Return storage track name.*
- void [SetStorageTrackName](#) (string name)  
*Update storage track name.*
- ostream & [Write](#) (ostream &stream) const  
*Write object to a stream.*
- istream & [Read](#) (istream &stream, const [CabNameMap](#) cabs)  
*Read an object from a stream.*

## Private Attributes

- double [layover](#)  
*The layover time.*
- int [stationindex](#)  
*The station index.*
- [Cab](#) \* [cab](#)  
*The [Cab](#) object.*
- vector< int > [notes](#)  
*The vector of note numbers.*
- [FlagType](#) [flag](#)  
*The type of stop, originating, terminating, or passing through.*
- string [storageTrackName](#)  
*The storage track name.*

### 9.162.1 Detailed Description

This class implements a stop.

This specifies the station the train goes through, even if it does not actually stop. A layover of 0 means the train does not stop and this station is a timekeeping check point.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.162.2 Member Enumeration Documentation

#### 9.162.2.1 FlagType

```
enum TTSupport::Stop::FlagType
```

Type of stop.

Origin for originating trains, Terminate for terminating trains, and Transit for trains passing through.

#### Enumerator

Origin	
Terminate	
Transit	

### 9.162.3 Constructor & Destructor Documentation

#### 9.162.3.1 Stop() [1/2]

```
TTSupport::Stop::Stop (
    int stationindex_ = 0,
    FlagType flag_ = Origin ) [inline]
```

Constructor: create a new stop.

##### Parameters

<i>stationindex_</i>	The index of the station.
<i>flag_</i>	The type of stop (originating, terminating, or passing through).

References cab, flag, layover, stationindex, and storageTrackName.

#### 9.162.3.2 Stop() [2/2]

```
TTSupport::Stop::Stop (
    const Stop & other ) [inline]
```

Copy constructor, create a stop from another stop.

##### Parameters

<i>other</i>	The other stop.
--------------	-----------------

References cab, flag, layover, notes, stationindex, and storageTrackName.

#### 9.162.3.3 ~Stop()

```
TTSupport::Stop::~Stop ( ) [inline]
```

Destructor.

## 9.162.4 Member Function Documentation

### 9.162.4.1 AddNote()

```
void TTSupport::Stop::AddNote (
    int note ) [inline]
```

Add a note.

#### Parameters

<i>note</i>	The note number.
-------------	------------------

References [i](#), and [notes](#).

### 9.162.4.2 Departure()

```
double TTSupport::Stop::Departure (
    double arrival ) const [inline]
```

Return departure time.

This is just the layover period added to the arrival time.

#### Parameters

<i>arrival</i>	The arrival time.
----------------	-------------------

References [layover](#).

### 9.162.4.3 Flag()

```
FlagType TTSupport::Stop::Flag ( ) const [inline]
```

Return the flag.

References [flag](#).

#### 9.162.4.4 Layover()

```
double TTSupport::Stop::Layover ( ) const [inline]
```

Return layover period.

References layover.

#### 9.162.4.5 Note()

```
int TTSupport::Stop::Note (
    int i ) const [inline]
```

Return the ith note.

Returns -1 if the index is out of range.

##### Parameters

<i>i</i>	The index of the note.
----------	------------------------

References i, and notes.

#### 9.162.4.6 NumberOfNotes()

```
int TTSupport::Stop::NumberOfNotes ( ) const [inline]
```

Return the number of notes.

References notes.

#### 9.162.4.7 operator=()

```
Stop& TTSupport::Stop::operator= (
    const Stop & other ) [inline]
```

Assignment operator, assign one stop to another stop.

## Parameters

<i>other</i>	The other stop.
--------------	-----------------

References cab, flag, layover, notes, stationindex, and storageTrackName.

## 9.162.4.8 Read()

```
istream& TTSupport::Stop::Read (  
    istream & stream,  
    const CabNameMap cabs )
```

Read an object from a stream.

## Parameters

<i>stream</i>	Stream to read from.
<i>cabs</i>	Map of cab names.

Referenced by SetStorageTrackName(), and TTSupport::Train::StartSMile().

## 9.162.4.9 RemoveNote()

```
void TTSupport::Stop::RemoveNote (  
    int note ) [inline]
```

Remove note.

## Parameters

<i>note</i>	The note number to remove.
-------------	----------------------------

References notes.

## 9.162.4.10 SetCab()

```
void TTSupport::Stop::SetCab (  
    Cab * newcab ) [inline]
```

Update the cab.

**Parameters**

<i>newcab</i>	The new cab value.
---------------	--------------------

References cab.

**9.162.4.11 SetLayover()**

```
void TTSupport::Stop::SetLayover (
    double period ) [inline]
```

Update layover period.

**Parameters**

<i>period</i>	New layover period.
---------------	---------------------

References layover.

**9.162.4.12 SetStorageTrackName()**

```
void TTSupport::Stop::SetStorageTrackName (
    string name ) [inline]
```

Update storage track name.

**Parameters**

<i>name</i>	The name of the storage track.
-------------	--------------------------------

References Read(), storageTrackName, lcc::stream, and Write().

Referenced by StorageTrackName().

**9.162.4.13 StationIndex()**

```
int TTSupport::Stop::StationIndex ( ) const [inline]
```

Return the station index.

References stationindex.



#### 9.162.4.14 StorageTrackName()

```
const char* TTSupport::Stop::StorageTrackName ( ) const [inline]
```

Return storage track name.

References SetStorageTrackName(), and storageTrackName.

#### 9.162.4.15 TheCab()

```
Cab* TTSupport::Stop::TheCab ( ) const [inline]
```

Return the cab.

References cab.

#### 9.162.4.16 Write()

```
ostream& TTSupport::Stop::Write (
    ostream & stream ) const
```

Write object to a stream.

##### Parameters

<i>stream</i>	Stream to write to.
---------------	---------------------

Referenced by SetStorageTrackName(), and TTSupport::Train::StartSMile().

### 9.162.5 Member Data Documentation

#### 9.162.5.1 cab

```
Cab* TTSupport::Stop::cab [private]
```

The Cab object.

Referenced by operator=(), TTSupport::Train::RemoveNoteFromTrain(), SetCab(), Stop(), and TheCab().

#### 9.162.5.2 flag

```
FlagType TTSupport::Stop::flag [private]
```

The type of stop, originating, terminating, or passing through.

Referenced by Flag(), operator=(), and Stop().

#### 9.162.5.3 layover

```
double TTSupport::Stop::layover [private]
```

The layover time.

Referenced by Departure(), Layover(), operator=(), TTSupport::Train::RemoveNoteFromTrain(), SetLayover(), and Stop().

#### 9.162.5.4 notes

```
vector<int> TTSupport::Stop::notes [private]
```

The vector of note numbers.

Referenced by AddNote(), TTSupport::Train::AddNoteToTrain(), Note(), TTSupport::Train::Note(), NumberOfNotes(), TTSupport::Train::NumberOfNotes(), operator=(), RemoveNote(), TTSupport::Train::RemoveNoteFromTrain(), and Stop().

#### 9.162.5.5 stationindex

```
int TTSupport::Stop::stationindex [private]
```

The station index.

Referenced by operator=(), StationIndex(), and Stop().

#### 9.162.5.6 storageTrackName

```
string TTSupport::Stop::storageTrackName [private]
```

The storage track name.

Referenced by operator=(), SetStorageTrackName(), Stop(), and StorageTrackName().

## 9.163 TTSupport::StorageTrack Class Reference

The [StorageTrack](#) class implements a storage track.

```
#include <Station.h>
```

### Public Member Functions

- [StorageTrack](#) (string name\_="Track 0")  
*Construct a storage track.*
- [~StorageTrack](#) ()  
*Destructor.*
- const char \* [Name](#) () const  
*Return the name of the storage track.*
- void [SetName](#) (string name\_)  
*Set the storage track's name.*
- const [Occupied](#) \* [IncludesTime](#) (double time) const  
*Return the occupation that includes the specified time;.*
- const [Occupied](#) \* [StoreTrain](#) (string train, double from, double to, string train2)  
*Insert train onto storage track for a time.*
- bool [RemovedStoredTrain](#) (double from, double to)  
*Remove stored train.*
- bool [UsedTimeRange](#) (double from, double to) const  
*Return true if the time range is in use.*
- const [Occupied](#) \* [FindOccupied](#) (double from, double to) const  
*Return occupation structure for a given time tange.*
- const [Occupied](#) \* [UpdateStoredTrain](#) (double from, double to, string train)  
*Replace a stored arrival train.*
- const [Occupied](#) \* [UpdateStoredTrain2](#) (double from, double to, string train)  
*Replace a stored departure train.*
- const [Occupied](#) \* [UpdateStoredTrainArrival](#) (double from, double to, double newArrival)  
*Update a train's arrival time.*
- const [Occupied](#) \* [UpdateStoredTrainDeparture](#) (double from, double to, double newDeparture)  
*Update a train's departure time.*
- [StorageTrack](#) (const [StorageTrack](#) &other)  
*Clone a [StorageTrack](#) – copy constructor.*
- [StorageTrack](#) & [operator=](#) (const [StorageTrack](#) &other)  
*Assign a [StorageTrack](#) to another [StorageTrack](#).*
- ostream & [Write](#) (ostream &stream) const  
*Write method.*
- istream & [Read](#) (istream &stream)  
*Read Method.*
- OccupiedMap::const\_iterator [FirstOccupied](#) () const  
*Return a const iterator for the first occupation.*
- OccupiedMap::const\_iterator [LastOccupied](#) () const  
*Return a const iterator for the last occupation.*

## Private Attributes

- string `name`  
*Name of the storage track.*
- `OccupiedMap` `occupations`  
*Map of occupations.*

### 9.163.1 Detailed Description

The `StorageTrack` class implements a storage track.

Storage tracks store trains at stations. Each storage track can only store one train at a given time. No checks are made to determining if the track is actually long enough for the train.

Each storage track has a name.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.163.2 Constructor & Destructor Documentation

#### 9.163.2.1 `StorageTrack()` [1/2]

```
TTSupport::StorageTrack::StorageTrack (
    string name_ = "Track 0" ) [inline]
```

Construct a storage track.

The name of the track is initialized.

#### Parameters

<code>name</code> ↔	The name of the storage track.
—	

#### 9.163.2.2 `~StorageTrack()`

```
TTSupport::StorageTrack::~~StorageTrack ( ) [inline]
```

Destructor.

### 9.163.2.3 StorageTrack() [2/2]

```
TTSupport::StorageTrack::StorageTrack (
    const StorageTrack & other ) [inline]
```

Clone a [StorageTrack](#) – copy constructor.

#### Parameters

<i>other</i>	The other <a href="#">StorageTrack</a> .
--------------	--

References name, and occupations.

## 9.163.3 Member Function Documentation

### 9.163.3.1 FindOccupied()

```
const Occupied* TTSupport::StorageTrack::FindOccupied (
    double from,
    double to ) const [inline]
```

Return occupation structure for a given time tange.

#### Parameters

<i>from</i>	The arrival time.
<i>to</i>	The departure time.

### 9.163.3.2 FirstOccupied()

```
OccupiedMap::const_iterator TTSupport::StorageTrack::FirstOccupied ( ) const [inline]
```

Return a const iterator for the first occupation.

### 9.163.3.3 IncludesTime()

```
const Occupied* TTSupport::StorageTrack::IncludesTime (
    double time ) const
```

Return the occupation that includes the specified time;.

## Parameters

<i>time</i>	The time to check for.
-------------	------------------------

## 9.163.3.4 LastOccupied()

```
OccupiedMap::const_iterator TTSupport::StorageTrack::LastOccupied ( ) const [inline]
```

Return a const iterator for the last occupation.

## 9.163.3.5 Name()

```
const char* TTSupport::StorageTrack::Name ( ) const [inline]
```

Return the name of the storage track.

## 9.163.3.6 operator=()

```
StorageTrack& TTSupport::StorageTrack::operator= (
    const StorageTrack & other ) [inline]
```

Assign a [StorageTrack](#) to another [StorageTrack](#).

## Parameters

<i>other</i>	The other <a href="#">StorageTrack</a> .
--------------	--

References name, occupations, TTSupport::Occupied::Read(), lcc::stream, and TTSupport::Occupied::Write().

## 9.163.3.7 Read()

```
istream& TTSupport::StorageTrack::Read (
    istream & stream )
```

Read Method.

Read object from a stream.

## Parameters

<i>stream</i>	Stream to read from.
---------------	----------------------

## 9.163.3.8 RemovedStoredTrain()

```
bool TTSupport::StorageTrack::RemovedStoredTrain (
    double from,
    double to )
```

Remove stored train.

## Parameters

<i>from</i>	The arrival time.
<i>to</i>	The departure time.

## 9.163.3.9 SetName()

```
void TTSupport::StorageTrack::SetName (
    string name_ ) [inline]
```

Set the storage track's name.

## Parameters

<i>name_</i>	The new name of the storage track.
--------------	------------------------------------

References TTSupport::Occupied::from.

## 9.163.3.10 StoreTrain()

```
const Occupied* TTSupport::StorageTrack::StoreTrain (
    string train,
    double from,
    double to,
    string train2 )
```

Insert train onto storage track for a time.

**Parameters**

<i>train</i>	The arriving train.
<i>from</i>	The arrival time.
<i>to</i>	The departure time.
<i>train2</i>	The departing train.

**9.163.3.11 UpdateStoredTrain()**

```
const Occupied* TTSupport::StorageTrack::UpdateStoredTrain (
    double from,
    double to,
    string train )
```

Replace a stored arrival train.

**Parameters**

<i>from</i>	The arrival time.
<i>to</i>	The departure time.
<i>train</i>	The new arriving train.

**9.163.3.12 UpdateStoredTrain2()**

```
const Occupied* TTSupport::StorageTrack::UpdateStoredTrain2 (
    double from,
    double to,
    string train )
```

Replace a stored departure train.

**Parameters**

<i>from</i>	The arrival time.
<i>to</i>	The departure time.
<i>train</i>	The new departing train.



#### 9.163.3.13 UpdateStoredTrainArrival()

```
const Occupied* TTSupport::StorageTrack::UpdateStoredTrainArrival (
    double from,
    double to,
    double newArrival )
```

Update a train's arrival time.

##### Parameters

<i>from</i>	The arrival time.
<i>to</i>	The departure time.
<i>newArrival</i>	The new arrival time.

#### 9.163.3.14 UpdateStoredTrainDeparture()

```
const Occupied* TTSupport::StorageTrack::UpdateStoredTrainDeparture (
    double from,
    double to,
    double newDeparture )
```

Update a train's departure time.

##### Parameters

<i>from</i>	The arrival time.
<i>to</i>	The departure time.
<i>newDeparture</i>	The new departure time.

#### 9.163.3.15 UsedTimeRange()

```
bool TTSupport::StorageTrack::UsedTimeRange (
    double from,
    double to ) const
```

Return true if the time range is in use.

##### Parameters

<i>from</i>	The arrival time.
<i>to</i>	The departure time.

#### 9.163.3.16 Write()

```
ostream& TTSupport::StorageTrack::Write (  
    ostream & stream ) const
```

Write method.

Write object to a stream.

##### Parameters

<i>stream</i>	Stream to write to.
---------------	---------------------

### 9.163.4 Member Data Documentation

#### 9.163.4.1 name

```
string TTSupport::StorageTrack::name [private]
```

Name of the storage track.

Referenced by operator=(), and StorageTrack().

#### 9.163.4.2 occupations

```
OccupiedMap TTSupport::StorageTrack::occupations [private]
```

Map of occupations.

Referenced by operator=(), and StorageTrack().

## 9.164 CTCTPanel::StraightBlock Class Reference

Straight Block object type.

## Public Member Functions

- [StraightBlock](#) (name, \_ctcpanel, \_canvas,...)  
*Construct a [StraightBlock](#) object.*
- [~StraightBlock](#) ()  
*Clean up all data objects and free up all resources.*
- [getv](#) ()  
*Method to get our value (state).*
- [setv](#) (value)  
*Method to set out value (state).*
- [geti](#) (ind)  
*Method to get the state of one of our indicators (none).*
- [seti](#) (ind, value)  
*Method to set an indicator's state (none).*
- [invoke](#) ()  
*Method to invoke the [StraightBlock](#).*

## Private Member Functions

- [\\_configureLabel](#) (option, value)  
*Method to update the label option.*

## Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*

### 9.164.1 Detailed Description

Straight Block object type.

These are on the schematic and represent a piece of track on the Schematic.

#### Parameters

<a href="#">_ctcpanel</a>	The <a href="#">CTCPanel</a> megawidget.
<a href="#">_canvas</a>	The schematic canvas to draw the <a href="#">StraightBlock</a> on.

## Parameters

...	<p>Options:</p> <ul style="list-style-type: none"> <li>• -x1 The first x coordinate of the object (readonly, default 0).</li> <li>• -y1 The first y coordinate of the object (readonly, default 0).</li> <li>• -x2 The second x coordinate of the object (readonly, default 0).</li> <li>• -y2 The second y coordinate of the object (readonly, default 0).</li> <li>• -controlpoint The name of the control point this label is part of (readonly, default MainLine).</li> <li>• -label The label of the <a href="#">StraightBlock</a> (default "").</li> <li>• -position The position of the label (readonly, default below).</li> <li>• -occupiedcommand A command to run to find out if the block is occupied (default {}).</li> </ul>
-----	--

Defined coords terminals:

- E1 First endpoint.
- E2 Second endpoint.

Defined values (states): none. Defined indicators: none.

## Author

Robert Heller <heller@deepsoft.com>

## 9.164.2 Constructor & Destructor Documentation

### 9.164.2.1 StraightBlock()

```
CTCPanel::StraightBlock::StraightBlock (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a [StraightBlock](#) object.

## Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the <a href="#">StraightBlock</a> on.
<code>...</code>	Option list.

**9.164.2.2** `~StraightBlock()`

```
CTCPanel::StraightBlock::~~StraightBlock ( )
```

Clean up all data objects and free up all resources.

**9.164.3** Member Function Documentation**9.164.3.1** `_configureLabel()`

```
CTCPanel::StraightBlock::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

**9.164.3.2** `geti()`

```
CTCPanel::StraightBlock::geti (
    ind )
```

Method to get the state of one of our indicators (none).

**9.164.3.3** `getv()`

```
CTCPanel::StraightBlock::getv ( )
```

Method to get our value (state).

**9.164.3.4 invoke()**

```
CTCPanel::StraightBlock::invoke ( )
```

Method to invoke the [StraightBlock](#).

**9.164.3.5 seti()**

```
CTCPanel::StraightBlock::seti (
    ind ,
    value )
```

Method to set an indicator's state (none).

**9.164.3.6 setv()**

```
CTCPanel::StraightBlock::setv (
    value )
```

Method to set out value (state).

**Parameters**

<i>value</i>	The new state to set.
--------------	-----------------------

**9.164.4 Member Data Documentation****9.164.4.1 canvas**

```
CTCPanel::StraightBlock::canvas [private]
```

The canvas component (parent widget component).

**9.164.4.2 ctcpnl**

```
CTCPanel::StraightBlock::ctcpnl [private]
```

The CTC Panel component (parent widget).

## 9.165 CTCPanel::StubYard Class Reference

Stub Yard object type.

### Public Member Functions

- [StubYard](#) (name, \_ctcpanel, \_canvas,...)  
*Construct a [StubYard](#) object.*
- [~StubYard](#) ()  
*Clean up all data objects and free up all resources.*
- [getv](#) ()  
*Method to get our value (state).*
- [setv](#) (value)  
*Method to set out value (state).*
- [geti](#) (ind)  
*Method to get the state of one of our indicators (none).*
- [seti](#) (ind, value)  
*Method to set an indicator's state (none).*
- [invoke](#) ()  
*Method to invoke the [StubYard](#).*

### Private Member Functions

- [\\_configureLabel](#) (option, value)  
*Method to update the label option.*

### Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*

### Static Private Attributes

- static [\\_StubYard\\_Poly](#)  
*Polygon coordinates for a stub yard.*

#### 9.165.1 Detailed Description

Stub Yard object type.

These are on the schematic and represent a piece of track on the Schematic.

## Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the <a href="#">StubYard</a> on.
<code>...</code>	Options: <ul style="list-style-type: none"> <li>• <code>-x</code> The x coordinate of the object (readonly, default 0).</li> <li>• <code>-y</code> The y coordinate of the object (readonly, default 0).</li> <li>• <code>-controlpoint</code> The name of the control point this label is part of (readonly, default Yard).</li> <li>• <code>-label</code> The label of the <a href="#">StubYard</a> (default "1").</li> <li>• <code>-position</code> The position of the label (readonly, default below).</li> <li>• <code>-orientation</code> The orientation (8-way) (readonly, default 0).</li> <li>• <code>-flipped</code> Whether the yard is flipped (readonly, default no).</li> <li>• <code>-occupiedcommand</code> A command to run to find out if the <a href="#">StubYard</a> is occupied (default {}).</li> </ul>

Defined coords terminals:

- Entry Yard throat.

Defined values (states): none. Defined indicators: none.

## Author

Robert Heller <heller@deepsoft.com>

## 9.165.2 Constructor & Destructor Documentation

### 9.165.2.1 StubYard()

```
CTCPanel::StubYard::StubYard (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a [StubYard](#) object.



## Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the <a href="#">StubYard</a> on.
<code>...</code>	Option list.

**9.165.2.2** `~StubYard()`

```
CTCPanel::StubYard::~~StubYard ( )
```

Clean up all data objects and free up all resources.

**9.165.3** Member Function Documentation**9.165.3.1** `_configureLabel()`

```
CTCPanel::StubYard::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

**9.165.3.2** `geti()`

```
CTCPanel::StubYard::geti (
    ind )
```

Method to get the state of one of our indicators (none).

**9.165.3.3** `getv()`

```
CTCPanel::StubYard::getv ( )
```

Method to get our value (state).

#### 9.165.3.4 invoke()

```
CTCPanel::StubYard::invoke ( )
```

Method to invoke the [StubYard](#).

#### 9.165.3.5 seti()

```
CTCPanel::StubYard::seti (
    ind ,
    value )
```

Method to set an indicator's state (none).

#### 9.165.3.6 setv()

```
CTCPanel::StubYard::setv (
    value )
```

Method to set out value (state).

##### Parameters

<i>value</i>	The new state to set.
--------------	-----------------------

### 9.165.4 Member Data Documentation

#### 9.165.4.1 \_StubYard\_Poly

```
CTCPanel::StubYard::_StubYard_Poly [static], [private]
```

Polygon coordinates for a stub yard.

## 9.165.4.2 canvas

```
CTCPanel::StubYard::canvas [private]
```

The canvas component (parent widget component).

## 9.165.4.3 ctcpnl

```
CTCPanel::StubYard::ctcpnl [private]
```

The CTC Panel component (parent widget).

## 9.166 CTCPanel::Switch Class Reference

Switch (turnout) object type.

### Public Member Functions

- [Switch](#) (name, \_ctcpnl, \_canvas,...)  
*Construct a [Switch](#) object.*
- [~Switch](#) ()  
*Clean up all data objects and free up all resources.*
- [getv](#) ()  
*Method to get our value (state).*
- [setv](#) (value)  
*Method to set out value (state).*
- [geti](#) (ind)  
*Method to get the state of one of our indicators (none).*
- [seti](#) (ind, value)  
*Method to set an indicator's state (none).*
- [invoke](#) ()  
*Method to invoke the switch.*

### Private Member Functions

- [\\_configureLabel](#) (option, value)  
*Method to update the label option.*

## Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*
- [state](#)  
*The state of the points.*

### 9.166.1 Detailed Description

Switch (turnout) object type.

These are on the schematic and represent a switch on the Schematic.

#### Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the switch on.
<code>...</code>	Options: <ul style="list-style-type: none"> <li>• <code>-x</code> The x coordinate of the object (readonly, default 0).</li> <li>• <code>-y</code> The y coordinate of the object (readonly, default 0).</li> <li>• <code>-controlpoint</code> The name of the control point this label is part of (readonly, default CP1).</li> <li>• <code>-label</code> The label of the switch (default "1").</li> <li>• <code>-orientation</code> The orientation (8-way) of the switch (readonly, default 0).</li> <li>• <code>-flipped</code> Whether or not the switch is flipped (readonly, default no).</li> <li>• <code>-statecommand</code> A command to run to get the switch's state (default {}).</li> <li>• <code>-occupiedcommand</code> A command to run to find out if the switch is occupied (default {}).</li> </ul>

Defined coords terminals:

- Common Common terminal (point end of switch).
- Main Mainline terminal.
- Divergence Branchline terminal.

Defined values (states):

- Normal Points are aligned for the mainline.
- Reverse Points are aligned for the branchline.
- Unknown Point are not aligned for any route (eg points are in motion).

Defined indicators: none.

**Author**

Robert Heller <heller@deepsoft.com>

**9.166.2 Constructor & Destructor Documentation****9.166.2.1 Switch()**

```
CTCPanel::Switch::Switch (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a [Switch](#) object.

**Parameters**

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the <a href="#">Switch</a> on.
<code>...</code>	Option list.

**9.166.2.2 ~Switch()**

```
CTCPanel::Switch::~~Switch ( )
```

Clean up all data objects and free up all resources.

**9.166.3 Member Function Documentation****9.166.3.1 \_configureLabel()**

```
CTCPanel::Switch::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

**9.166.3.2   geti()**

```
CTCPanel::Switch::geti (
    ind )
```

Method to get the state of one of our indicators (none).

**9.166.3.3   getv()**

```
CTCPanel::Switch::getv ( )
```

Method to get our value (state).

**9.166.3.4   invoke()**

```
CTCPanel::Switch::invoke ( )
```

Method to invoke the switch.

**9.166.3.5   seti()**

```
CTCPanel::Switch::seti (
    ind ,
    value )
```

Method to set an indicator's state (none).

**9.166.3.6   setv()**

```
CTCPanel::Switch::setv (
    value )
```

Method to set out value (state).

**Parameters**

<i>value</i>	The new state to set.
--------------	-----------------------

#### 9.166.4 Member Data Documentation

##### 9.166.4.1 canvas

CTCPanel::Switch::canvas [private]

The canvas component (parent widget component).

##### 9.166.4.2 ctcpnl

CTCPanel::Switch::ctcpnl [private]

The CTC Panel component (parent widget).

##### 9.166.4.3 state

CTCPanel::Switch::state [private]

The state of the points.

### 9.167 FCFSupport::SwitchList Class Reference

The global switch list structure.

```
#include <SwitchList.h>
```

## Public Member Functions

- [SwitchList](#) ()  
*Constructor.*
- [~SwitchList](#) ()  
*Destructor.*
- void [ResetSwitchList](#) ()  
*Reset the switch list pointer.*
- void [DiscardSwitchList](#) ()  
*Clobber the switch list.*
- void [AddSwitchListElement](#) (const [Industry](#) \*pickloc, const [Car](#) \*pickcar, const [Train](#) \*picktrain, const [Train](#) \*lasttrain, const [Industry](#) \*istop)  
*Add a switch list element for a manifest freight (industry stop).*
- void [AddSwitchListElement](#) (const [Industry](#) \*pickloc, const [Car](#) \*pickcar, const [Train](#) \*picktrain, const [Train](#) \*lasttrain, const [Station](#) \*ssstop)  
*Add a switch list element for a local freight (station stop).*
- [SwitchListElement](#) & [operator\[\]](#) (int ielement)  
*Random index access to the switch list.*
- const [SwitchListElement](#) [operator\[\]](#) (int ielement) const  
*Random index access to the switch list, const version.*
- int [NextSwitchListForCarAndIndustry](#) (const [Car](#) \*car, const [Industry](#) \*industry)  
*Return the next switch list list element for a selected car and industry.*
- unsigned int [PickIndex](#) () const  
*Return the pick index.*
- int [LimitCars](#) () const  
*Return the limit count.*
- void [ResetLastIndex](#) ()  
*Reset the last index.*
- bool [PickLocationEq](#) (int Gx, const [Industry](#) \*Ix) const  
*Is the selected element for the specificed industry?*
- bool [PickCarEq](#) (int Gx, const [Car](#) \*Cx) const  
*Is the selected element for the specificed car?*
- bool [PickTrainEq](#) (int Gx, const [Train](#) \*Tx) const  
*Is the selected element for the specificed train?*

## Private Attributes

- [SwitchListElementVector](#) theList  
*The switch list vector.*
- unsigned int [pickIndex](#)  
*The pick index.*
- int [limitCars](#)  
*The limit index.*
- int [lastIndex](#)  
*The last index.*



## Friends

- ostream & [operator<<](#) (ostream &stream, const [SwitchList](#) &list)  
*Output stream operator for switch lists.*

### 9.167.1 Detailed Description

The global switch list structure.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.167.2 Constructor & Destructor Documentation

#### 9.167.2.1 SwitchList()

```
FCFSupport::SwitchList::SwitchList ( )
```

Constructor.

#### 9.167.2.2 ~SwitchList()

```
FCFSupport::SwitchList::~~SwitchList ( )
```

Destructor.

### 9.167.3 Member Function Documentation

#### 9.167.3.1 AddSwitchListElement() [1/2]

```
void FCFSupport::SwitchList::AddSwitchListElement (
    const Industry * pickloc,
    const Car * pickcar,
    const Train * picktrain,
    const Train * lasttrain,
    const Industry * istop )
```

Add a switch list element for a manifest freight (industry stop).

## Parameters

<i>pickloc</i>	Pickup location of car.
<i>pickcar</i>	<a href="#">Car</a> being picked up by this train.
<i>picktrain</i>	<a href="#">Train</a> picking this car up.
<i>lasttrain</i>	The last train this car was on.
<i>istop</i>	Where this train will drop this car.

## 9.167.3.2 AddSwitchListElement() [2/2]

```
void FCFSupport::SwitchList::AddSwitchListElement (
    const Industry * pickloc,
    const Car * pickcar,
    const Train * picktrain,
    const Train * lasttrain,
    const Station * sstop )
```

Add a switch list element for a local freight (station stop).

## Parameters

<i>pickloc</i>	Pickup location of car.
<i>pickcar</i>	<a href="#">Car</a> being picked up by this train.
<i>picktrain</i>	<a href="#">Train</a> picking this car up.
<i>lasttrain</i>	The last train this car was on.
<i>sstop</i>	Where this train will drop this car.

## 9.167.3.3 DiscardSwitchList()

```
void FCFSupport::SwitchList::DiscardSwitchList ( )
```

Clobber the switch list.

## 9.167.3.4 LimitCars()

```
int FCFSupport::SwitchList::LimitCars ( ) const [inline]
```

Return the limit count.

#### 9.167.3.5 NextSwitchListForCarAndIndustry()

```
int FCFSupport::SwitchList::NextSwitchListForCarAndIndustry (
    const Car * car,
    const Industry * industry )
```

Return the next switch list list element for a selected car and industry.

##### Parameters

<i>car</i>	The selected car.
<i>industry</i>	The selected industry.

#### 9.167.3.6 operator[]() [1/2]

```
SwitchListElement& FCFSupport::SwitchList::operator[] (
    int ielement )
```

Random index access to the switch list.

##### Parameters

<i>ielement</i>	The index into the switch list.
-----------------	---------------------------------

#### 9.167.3.7 operator[]() [2/2]

```
const SwitchListElement FCFSupport::SwitchList::operator[] (
    int ielement ) const
```

Random index access to the switch list, const version.

##### Parameters

<i>ielement</i>	The index into the switch list.
-----------------	---------------------------------

#### 9.167.3.8 PickCarEq()

```
bool FCFSupport::SwitchList::PickCarEq (
    int Gx,
    const Car * Cx ) const
```

Is the selected element for the specified car?

**Parameters**

<i>Gx</i>	The index to check.
<i>Cx</i>	The car to check for.

**9.167.3.9 PickIndex()**

```
unsigned int FCFSupport::SwitchList::PickIndex ( ) const [inline]
```

Return the pick index.

**9.167.3.10 PickLocationEq()**

```
bool FCFSupport::SwitchList::PickLocationEq (
    int Gx,
    const Industry * Ix ) const
```

Is the selected element for the specified industry?

**Parameters**

<i>Gx</i>	The index to check.
<i>Ix</i>	The industry to check for.

**9.167.3.11 PickTrainEq()**

```
bool FCFSupport::SwitchList::PickTrainEq (
    int Gx,
    const Train * Tx ) const
```

Is the selected element for the specified train?

**Parameters**

<i>Gx</i>	The index to check.
<i>Tx</i>	The train to check for.

#### 9.167.3.12 ResetLastIndex()

```
void FCFSupport::SwitchList::ResetLastIndex ( ) [inline]
```

Reset the last index.

References FCFSupport::operator<<(), and lcc::stream.

#### 9.167.3.13 ResetSwitchList()

```
void FCFSupport::SwitchList::ResetSwitchList ( )
```

Reset the switch list pointer.

### 9.167.4 Friends And Related Function Documentation

#### 9.167.4.1 operator<<

```
ostream& operator<< (
    ostream & stream,
    const SwitchList & list ) [friend]
```

Output stream operator for switch lists.

##### Parameters

<i>stream</i>	The stream to write to.
<i>list</i>	The switch list to write out.

### 9.167.5 Member Data Documentation

#### 9.167.5.1 lastIndex

```
int FCFSupport::SwitchList::lastIndex [private]
```

The last index.

#### 9.167.5.2 limitCars

```
int FCFSupport::SwitchList::limitCars [private]
```

The limit index.

#### 9.167.5.3 pickIndex

```
unsigned int FCFSupport::SwitchList::pickIndex [private]
```

The pick index.

#### 9.167.5.4 theList

```
SwitchListElementVector FCFSupport::SwitchList::theList [private]
```

The switch list vector.

## 9.168 FCFSupport::SwitchListElement Class Reference

This class implements each switch list element.

```
#include <SwitchList.h>
```

### Classes

- union [StationOrIndustry](#)

*A const pointer to a train's stop, which can be either a station or an industry, depending on what kind of train it is.*

## Public Member Functions

- [SwitchListElement](#) ()  
*Default constructor.*
- [SwitchListElement](#) (const [SwitchListElement](#) &other)  
*Copy constructor.*
- [SwitchListElement](#) & operator= (const [SwitchListElement](#) &other)  
*Assignment operator.*
- [SwitchListElement](#) (const [Industry](#) \*pickloc, const [Car](#) \*pickcar, const [Train](#) \*picktrain, const [Train](#) \*lasttrain, const [Industry](#) \*istop)  
*Constructor, given a manifest freight's stop at an industry.*
- [SwitchListElement](#) (const [Industry](#) \*pickloc, const [Car](#) \*pickcar, const [Train](#) \*picktrain, const [Train](#) \*lasttrain, const [Station](#) \*sstop)  
*Constructor, given a local freight's stop at a station.*
- const [Industry](#) \* [PickLocation](#) () const  
*Return the pickup location for this switch list element.*
- const [Car](#) \* [PickCar](#) () const  
*Return the car picked up for this switch list element.*
- const [Train](#) \* [PickTrain](#) () const  
*Return the pickup train for this switch list element.*
- const [Train](#) \* [LastTrain](#) () const  
*Return the train train for the car this switch list element is for.*
- const [Industry](#) \* [DropStopIndustry](#) () const  
*Return the industry this switch list element is dropping off at.*
- const [Station](#) \* [DropStopStation](#) () const  
*Return the station this switch list element is dropping off at.*
- bool [DropStopEQ](#) (int Px) const  
*Is the drop stop at the stop number specified?*

## Private Attributes

- const [Industry](#) \* pickLoc  
*The pickup industry.*
- const [Car](#) \* pickCar  
*The car picked up.*
- const [Train](#) \* pickTrain  
*The train picking this car up.*
- const [Train](#) \* lastTrain  
*The train that previously handled this car.*
- [StationOrIndustry](#) dropStop  
*The station or industry where this car will be dropped off at.*

## Friends

- class [System](#)  
*The [System](#) class is a friend.*

### 9.168.1 Detailed Description

This class implements each switch list element.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.168.2 Constructor & Destructor Documentation

#### 9.168.2.1 SwitchListElement() [1/4]

```
FCFSupport::SwitchListElement::SwitchListElement ( ) [inline]
```

Default constructor.

Initialise all slots to NULL.

References dropStop, FCFSupport::SwitchListElement::StationOrIndustry::industry, lastTrain, pickCar, pickLoc, and pickTrain.

#### 9.168.2.2 SwitchListElement() [2/4]

```
FCFSupport::SwitchListElement::SwitchListElement (
    const SwitchListElement & other ) [inline]
```

Copy constructor.

#### Parameters

<i>other</i>	The other switch list element.
--------------	--------------------------------

References dropStop, FCFSupport::SwitchListElement::StationOrIndustry::industry, lastTrain, pickCar, pickLoc, and pickTrain.

#### 9.168.2.3 SwitchListElement() [3/4]

```
FCFSupport::SwitchListElement::SwitchListElement (
    const Industry * pickloc,
```



```

const Car * pickcar,
const Train * picktrain,
const Train * lasttrain,
const Industry * istop ) [inline]

```

Constructor, given a manifest freight's stop at an industry.

#### Parameters

<i>pickloc</i>	Pickup location of car.
<i>pickcar</i>	<a href="#">Car</a> being picked up by this train.
<i>picktrain</i>	<a href="#">Train</a> picking this car up.
<i>lasttrain</i>	The last train this car was on.
<i>istop</i>	Where this train will drop this car.

References [dropStop](#), [FCFSupport::SwitchListElement::StationOrIndustry::industry](#), [lastTrain](#), [pickCar](#), [pickLoc](#), and [pickTrain](#).

#### 9.168.2.4 SwitchListElement() [ 4 / 4 ]

```

FCFSupport::SwitchListElement::SwitchListElement (
    const Industry * pickloc,
    const Car * pickcar,
    const Train * picktrain,
    const Train * lasttrain,
    const Station * sstop ) [inline]

```

Constructor, given a local freight's stop at a station.

#### Parameters

<i>pickloc</i>	Pickup location of car.
<i>pickcar</i>	<a href="#">Car</a> being picked up by this train.
<i>picktrain</i>	<a href="#">Train</a> picking this car up.
<i>lasttrain</i>	The last train this car was on.
<i>sstop</i>	Where this train will drop this car.

References [dropStop](#), [lastTrain](#), [pickCar](#), [pickLoc](#), [pickTrain](#), and [FCFSupport::SwitchListElement::StationOrIndustry::station](#).

### 9.168.3 Member Function Documentation

### 9.168.3.1 DropStopEQ()

```
bool FCFSupport::SwitchListElement::DropStopEQ (
    int Px ) const [inline]
```

Is the drop stop at the stop number specified?

#### Parameters

<i>Px</i>	The train's stop number we are checking against.
-----------	--

References dropStop, FCFSupport::SwitchListElement::StationOrIndustry::industry, FCFSupport::Train::IndustryStop(), FCFSupport::Train::Manifest, pickTrain, FCFSupport::SwitchListElement::StationOrIndustry::station, FCFSupport::Train::StationStop(), and FCFSupport::Train::Type().

### 9.168.3.2 DropStopIndustry()

```
const Industry* FCFSupport::SwitchListElement::DropStopIndustry( ) const [inline]
```

Return the industry this switch list element is dropping off at.

References dropStop, FCFSupport::SwitchListElement::StationOrIndustry::industry, FCFSupport::Train::Manifest, pickTrain, and FCFSupport::Train::Type().

### 9.168.3.3 DropStopStation()

```
const Station* FCFSupport::SwitchListElement::DropStopStation ( ) const [inline]
```

Return the station this switch list element is dropping off at.

References dropStop, FCFSupport::SwitchListElement::StationOrIndustry::industry, FCFSupport::Train::Manifest, FCFSupport::Industry::MyStation(), pickTrain, FCFSupport::SwitchListElement::StationOrIndustry::station, and FCFSupport::Train::Type().

### 9.168.3.4 LastTrain()

```
const Train* FCFSupport::SwitchListElement::LastTrain ( ) const [inline]
```

Return the train train for the car this switch list element is for.

References lastTrain.

### 9.168.3.5 operator=()

```
SwitchListElement& FCFSupport::SwitchListElement::operator= (
    const SwitchListElement & other ) [inline]
```

Assignment operator.

## Parameters

<i>other</i>	The other switch list element.
--------------	--------------------------------

References dropStop, FCFSupport::SwitchListElement::StationOrIndustry::industry, lastTrain, pickCar, pickLoc, and pickTrain.

### 9.168.3.6 PickCar()

```
const Car* FCFSupport::SwitchListElement::PickCar ( ) const [inline]
```

Return the car picked up for this switch list element.

References pickCar.

### 9.168.3.7 PickLocation()

```
const Industry* FCFSupport::SwitchListElement::PickLocation ( ) const [inline]
```

Return the pickup location for this switch list element.

References pickLoc.

### 9.168.3.8 PickTrain()

```
const Train* FCFSupport::SwitchListElement::PickTrain ( ) const [inline]
```

Return the pickup train for this switch list element.

References pickTrain.

## 9.168.4 Friends And Related Function Documentation

### 9.168.4.1 System

```
friend class System [friend]
```

The [System](#) class is a friend.

## 9.168.5 Member Data Documentation

### 9.168.5.1 dropStop

`StationOrIndustry` FCFSupport::SwitchListElement::dropStop [private]

The station or industry where this car will be dropped off at.

Referenced by DropStopEQ(), DropStopIndustry(), DropStopStation(), operator=(), and SwitchListElement().

### 9.168.5.2 lastTrain

`const Train*` FCFSupport::SwitchListElement::lastTrain [private]

The train that previously handled this car.

Referenced by LastTrain(), operator=(), and SwitchListElement().

### 9.168.5.3 pickCar

`const Car*` FCFSupport::SwitchListElement::pickCar [private]

The car picked up.

Referenced by operator=(), PickCar(), and SwitchListElement().

### 9.168.5.4 pickLoc

`const Industry*` FCFSupport::SwitchListElement::pickLoc [private]

The pickup industry.

Referenced by operator=(), PickLocation(), and SwitchListElement().

## 9.168.5.5 pickTrain

```
const Train* FCFSupport::SwitchListElement::pickTrain [private]
```

The train picking this car up.

Referenced by DropStopEQ(), DropStopIndustry(), DropStopStation(), operator=(), PickTrain(), and SwitchListElement().

## 9.169 CTCPanel::SWPlate Class Reference

Switch plate object type.

### Public Member Functions

- [SWPlate](#) (name, \_ctcpanel, \_canvas,...)  
*Construct a [SWPlate](#) object.*
- [~SWPlate](#) ()  
*Clean up all data objects and free up all resources.*
- [getv](#) ()  
*Method to get our value (lever position).*
- [setv](#) (state)  
*Method to set out value (level position).*
- [geti](#) (ind)  
*Method to get the state of one of our indicators.*
- [seti](#) (ind, value)  
*Method to set an indicator's state.*
- [invoke](#) ()  
*Method to invoke the switch plate.*

### Private Member Functions

- [\\_configureLabel](#) (option, value)  
*Method to update the label option.*

### Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*

## Static Private Attributes

- static [\\_PlatePolygon](#)

*Polygon coordinates for the plate.*

### 9.169.1 Detailed Description

Switch plate object type.

These are on the control panel and represent levers for controlling track switches (aka turnouts). They have a lever that can be in two positions, normal (switch aligned for the main route) and reversed (switch aligned for the divergent route).

#### Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The control panel canvas to draw the switch plate on.
<code>...</code>	Options: <ul style="list-style-type: none"> <li>• <code>-x</code> The x coordinate of the object (readonly, default 0).</li> <li>• <code>-y</code> The y coordinate of the object (readonly, default 0).</li> <li>• <code>-label</code> The label of the switch plate (default 1).</li> <li>• <code>-controlpoint</code> The name of the control point this switch is part of (readonly, default CP1).</li> <li>• <code>-normalcommand</code> The Tcl script to run when switch is set to normal (default {}). </li> <li>• <code>-reversecommand</code> The Tcl script to run when switch is set to reverse (default {}). </li> </ul>

Defined coords terminals:

- `xy` The base coords of the object.

Defined values (states):

- N Normal.
- R Reversed.

Defined indicators:

- N Normal indicator (green if on).
- R Reversed indicator (yellow if on).

#### Author

Robert Heller <heller@deepsoft.com>

## 9.169.2 Constructor & Destructor Documentation

### 9.169.2.1 SWPlate()

```
CTCPanel::SWPlate::SWPlate (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a [SWPlate](#) object.

#### Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The control panel canvas to draw the switch plate on.
<code>...</code>	Option list.

### 9.169.2.2 ~SWPlate()

```
CTCPanel::SWPlate::~~SWPlate ( )
```

Clean up all data objects and free up all resources.

## 9.169.3 Member Function Documentation

### 9.169.3.1 \_configureLabel()

```
CTCPanel::SWPlate::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

### 9.169.3.2 geti()

```
CTCPanel::SWPlate::geti (
    ind )
```

Method to get the state of one of our indicators.

**Parameters**

<i>ind</i>	The indicator to fetch state information for.
------------	---

**9.169.3.3   getv()**

```
CTCPanel::SWPlate::getv ( )
```

Method to get our value (lever position).

**9.169.3.4   invoke()**

```
CTCPanel::SWPlate::invoke ( )
```

Method to invoke the switch plate.

One of the command scripts is executed depending on the lever position.

**9.169.3.5   seti()**

```
CTCPanel::SWPlate::seti (
    ind ,
    value )
```

Method to set an indicator's state.

**Parameters**

<i>ind</i>	The indicator to set.
<i>value</i>	The state to set it to.

**9.169.3.6   setv()**

```
CTCPanel::SWPlate::setv (
    state )
```

Method to set out value (level position).



**Parameters**

<i>state</i>	The new state to set.
--------------	-----------------------

**9.169.4 Member Data Documentation****9.169.4.1 \_PlatePolygon**

CTCPanel::SWPlate::\_PlatePolygon [static], [private]

Polygon coordinates for the plate.

**9.169.4.2 canvas**

CTCPanel::SWPlate::canvas [private]

The canvas component (parent widget component).

**9.169.4.3 ctcpnl**

CTCPanel::SWPlate::ctcpnl [private]

The CTC Panel component (parent widget).

**9.170 FCFSupport::System Class Reference**

This is the main Freight [Car](#) Forwarder class.

```
#include <System.h>
```

**Public Types**

- enum [CarTypeReport](#) { [All](#), [Type](#), [Summary](#) }  
*Types of car type reports.*
- enum [CarLocationType](#) { [INDUSTRY](#), [STATION](#), [DIVISION](#), [ALL](#) }  
*Types of location report.*

## Public Member Functions

- [System](#) (const char \*systemfile, int seed, char \*\*outmessage=NULL)  
*The constructor for the system.*
- [~System](#) ()  
*The destructor frees all memory and generally cleans things up.*
- const char \* [SystemName](#) () const  
*Return the system name.*
- const char \* [SystemFile](#) () const  
*Return the system file's full path name.*
- const char \* [IndustriesFile](#) () const  
*Return the industry file's full path name.*
- const char \* [TrainsFile](#) () const  
*Return the trains file's full path name.*
- const char \* [OrdersFile](#) () const  
*Return the train orders file's full path name.*
- const char \* [OwnersFile](#) () const  
*Return the Owners file's full path name.*
- const char \* [CarTypesFile](#) () const  
*Return the [Car](#) Types file's full path name.*
- const char \* [CarsFile](#) () const  
*Return the Cars file's full path name.*
- const char \* [StatsFile](#) () const  
*Return the Statistics file's full path name.*
- int [NumberOfDivisions](#) () const  
*return the number of divisions loaded.*
- const [Division](#) \* [FindDivisionByIndex](#) (int i) const  
*Find a division by its index.*
- const [Division](#) \* [FindDivisionBySymbol](#) (char symbol) const  
*Find a division by its symbol.*
- [FCFSupport::Division](#) \* [TheDivision](#) (int i)  
*[Division](#) indexing function.*
- int [NumberOfStations](#) () const  
*The number of stations loaded.*
- [FCFSupport::Station](#) \* [TheStation](#) (int i)  
*[Station](#) indexing function.*
- const [FCFSupport::Station](#) \* [FindStationByName](#) (string name, string comment) const  
*[Station](#) indexing (by name) function.*
- int [NumberOfTrains](#) () const  
*The number of trains loaded.*
- [Train](#) \* [TrainByIndex](#) (int i)  
*[Train](#) indexing function.*
- const [Train](#) \* [FindTrainByIndex](#) (int i) const  
*Find a train by its index.*
- [Train](#) \* [TrainByName](#) (const char \*name)  
*[Train](#) indexing (by name) function.*
- const [Train](#) \* [FindTrainByName](#) (const char \*name) const

- Find a train by its name.*

  - int [NumberOfIndustries](#) () const

*Return the number of industries loaded.*
- [FCFSupport::Industry](#) \* [TheIndustry](#) (int i)

*Industry indexing function.*
- const [FCFSupport::Industry](#) \* [FindIndustryByIndex](#) (int i) const

*Find an industry by its index.*
- const [FCFSupport::Industry](#) \* [FindIndustryByName](#) (string name) const

*Find an industry by its name.*
- char [CarTypesOrder](#) (int i) const

*Access a car type by index.*
- int [CarTypesOrderIndex](#) (char type) const

*Car type order index.*
- [CarType](#) \* [TheCarType](#) (char c)

*Get a car type class instance pointer given a car type.*
- [CarGroup](#) \* [TheCarGroup](#) (int i) const

*Get a car class instance pointer given a car group index.*
- int [NumberOfCars](#) () const
- [Owner](#) \* [TheOwner](#) (const char \*initials)

*Get a car owner class instance pointer given a car owner's initials.*
- void [AddOwner](#) (const char \*initials)

*Create a new owner given a set of initials.*
- [FCFSupport::Car](#) \* [TheCar](#) (int i) const

*Get a car by index.*
- void [AddCar](#) ([FCFSupport::Car](#) \*newcar)

*Add a new car to the array of cars.*
- int [SessionNumber](#) () const

*Return the session number.*
- int [ShiftNumber](#) () const

*Return the shift number.*
- int [TotalShifts](#) () const

*Return the total number of shifts.*
- int [NextShift](#) ()

*Increment the shift number.*
- int [TotalCars](#) () const

*Return the total number of cars.*
- int [RanAllTrains](#) () const

*Ran all trains?*
- void [DeleteAllExistingCars](#) ()

*Delete all existing cars.*
- bool [LoadCarFile](#) (char \*\*outmessage=NULL)

*(Re-)Load the car file.*
- bool [LoadStatsFile](#) (char \*\*outmessage=NULL)

*Load the stats file.*
- bool [SaveCars](#) (char \*\*outmessage=NULL)

*Save cars (and stats).*
- const [FCFSupport::Industry](#) \* [IndScrapYard](#) () const

- Return a pointer to the scrap yard.*

  - int [StatsPeriod](#) () const

*Return the current stats period.*
- int [TrainIndex](#) (const [FCFSupport::Train](#) \*train) const

*Return a train's index.*
- int [IndustryIndex](#) (const [FCFSupport::Industry](#) \*indus) const

*Return an industry's index.*
- int [CarsMoved](#) () const

*Return the number of cars moved.*
- int [CarsAtDest](#) () const

*Return the number of cars that are at their destinations.*
- int [CarsNotMoved](#) () const

*Return the number of cars not moved at all.*
- int [CarsMovedOnce](#) () const

*Return the number of cars moved once.*
- int [CarsMovedTwice](#) () const

*Return the number of cars moved twice.*
- int [CarsMovedThree](#) () const

*Return the number of cars moved three times.*
- int [CarsMovedMore](#) () const

*Return the number of cars moved more than three times.*
- int [CarMovements](#) () const

*Return the total number of car movements.*
- int [CarsInTransit](#) () const

*Return the number of cars still in transit.*
- int [CarsAtWorkBench](#) () const

*Return the number of cars on the RIP track (the workbench).*
- int [CarsAtDest\\_CarsInTransit](#) () const

*Return the number of cars at their destinations plus the number of cars in transit.*
- bool [PrintYards](#) () const

*Print yard lists flag.*
- void [SetPrintYards](#) (bool flag)

*Set the print yard lists flag.*
- bool [PrintAlpha](#) () const

*Print the alphabetical listing flag.*
- void [SetPrintAlpha](#) (bool flag)

*Set the print alphabetical listing flag.*
- bool [PrintAtwice](#) () const

*Print second copy of the alphabetical listing flag.*
- void [SetPrintAtwice](#) (bool flag)

*Set the print second copy of the alphabetical listing flag.*
- bool [PrintList](#) () const

*Print the switch list order flag.*
- void [SetPrintList](#) (bool flag)

*Set the print switch list order flag.*
- bool [PrintLtwice](#) () const

*Print a second copy of the switch list order flag.*

- void [SetPrintLtwice](#) (bool flag)  
*Set the print a second copy of the switch list order flag.*
- bool [PrintDispatch](#) () const  
*Print dispatcher report sheet.*
- void [SetPrintDispatch](#) (bool flag)  
*Set the print dispatcher report sheet.*
- bool [Printem](#) () const  
*Print train enroute switch list.*
- void [SetPrintem](#) (bool flag)  
*Set the print train enroute switch list.*
- const [FCFSupport::Industry](#) \* [IndRipTrack](#) ()  
*Return a pointer to the RIP track (workbench).*
- const [FCFSupport::Industry](#) \* [IndRipTrackConst](#) () const  
*Const version of the pointer to the RIP track (workbench).*
- void [RestartLoop](#) ()  
*Reset loop variables.*
- void [Randomize](#) (int seed)  
*Set the random seed.*
- double [Random](#) ()  
*Return a random number between 0.0 and 1.0.*
- void [CarAssignment](#) (const [FCFSupport::WorkInProgressCallback](#) \*WIP, const [FCFSupport::LogMessageCallback](#) \*Log, const [FCFSupport::ShowBannerCallback](#) \*banner, char \*\*outmessage=NULL)  
*Car assignment procedure.*
- void [RunAllTrains](#) (const [FCFSupport::WorkInProgressCallback](#) \*WIP, const [FCFSupport::LogMessageCallback](#) \*Log, const [FCFSupport::ShowBannerCallback](#) \*banner, [FCFSupport::PrinterDevice](#) \*printer, const [FCFSupport::TrainDisplayCallback](#) \*traindisplay)  
*Run all trains procedure.*
- void [RunBoxMoves](#) (const [FCFSupport::WorkInProgressCallback](#) \*WIP, const [FCFSupport::LogMessageCallback](#) \*Log, const [FCFSupport::ShowBannerCallback](#) \*banner, [FCFSupport::PrinterDevice](#) \*printer, const [FCFSupport::TrainDisplayCallback](#) \*traindisplay)  
*Run all boxmove trains.*
- void [PrintAllLists](#) (const [FCFSupport::LogMessageCallback](#) \*Log, const [FCFSupport::ShowBannerCallback](#) \*banner, [FCFSupport::PrinterDevice](#) \*printer)  
*Print all of the various yard and switch lists.*
- void [RunOneTrain](#) ([Train](#) \*train, bool boxMove, const [FCFSupport::TrainDisplayCallback](#) \*traindisplay, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer)  
*Run one single train.*
- void [ShowCarsNotMoved](#) (const [FCFSupport::LogMessageCallback](#) \*Log, const [FCFSupport::ShowBannerCallback](#) \*banner) const  
*Display cars not moved.*
- void [ShowCarMovements](#) (bool showAll, const [FCFSupport::Industry](#) \*Ix, const [FCFSupport::Train](#) \*Tx, const [FCFSupport::LogMessageCallback](#) \*Log, const [FCFSupport::ShowBannerCallback](#) \*banner) const  
*Show all car movements.*
- void [ShowTrainCars](#) (const [Train](#) \*Tx, const [FCFSupport::LogMessageCallback](#) \*Log, const [FCFSupport::ShowBannerCallback](#) \*banner) const  
*Show cars moved by a specific train.*
- void [ShowCarsInDivision](#) (const [Division](#) \*division, const [FCFSupport::LogMessageCallback](#) \*Log, const [FCFSupport::ShowBannerCallback](#) \*banner) const

- Show cars in a specified division.*

  - void [ShowTrainTotals](#) (const [FCFSupport::LogMessageCallback](#) \*Log, const [FCFSupport::ShowBannerCallback](#) \*banner) const
- Show train totals.*

  - void [ShowUnassignedCars](#) (const [FCFSupport::LogMessageCallback](#) \*Log, const [FCFSupport::ShowBannerCallback](#) \*banner) const
- Show unassigned cars.*

  - void [ReloadCarFile](#) (char \*\*outmessage)
- Reload car file.*

  - void [ResetIndustryStats](#) ()
- Reset industry statistics.*

  - void [ReportIndustries](#) (const [FCFSupport::WorkInProgressCallback](#) \*WIP, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer, char \*\*outmessage=NULL) const
- Report on all industries.*

  - void [ReportTrains](#) (const [FCFSupport::WorkInProgressCallback](#) \*WIP, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer, char \*\*outmessage=NULL) const
- Report on all trains.*

  - void [ReportCars](#) (const [FCFSupport::WorkInProgressCallback](#) \*WIP, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer, char \*\*outmessage=NULL) const
- Report on all cars.*

  - void [ReportCarsNotMoved](#) (const [FCFSupport::WorkInProgressCallback](#) \*WIP, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer, char \*\*outmessage=NULL) const
- Report on cars not moved.*

  - void [ReportCarTypes](#) ([CarTypeReport](#) rtype, char carType, [FCFSupport::PrinterDevice](#) \*printer, char \*\*outmessage=NULL) const
- Report on car types.*

  - void [ReportCarLocations](#) ([CarLocationType](#) cltype, int index, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer, char \*\*outmessage=NULL)
- Car location report.*

  - void [ReportAnalysis](#) (const [FCFSupport::WorkInProgressCallback](#) \*WIP, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer, char \*\*outmessage=NULL) const
- Industry analysis report.*

  - void [ReportCarOwners](#) (string ownerInitials, const [FCFSupport::WorkInProgressCallback](#) \*WIP, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer, char \*\*outmessage=NULL) const
- Report on a specified car owner.*

  - int [FindIndustryIndex](#) (const [FCFSupport::Industry](#) \*industry) const
- Find an industry's index.*

  - int [FindStationIndex](#) (const [FCFSupport::Station](#) \*station) const
- Find a station's index.*

  - int [FindDivisionIndex](#) (const [FCFSupport::Division](#) \*division) const
- Find a division's index.*

  - void [GetCarStatus](#) (const [Car](#) \*car, string &status, string &carTypeDescr) const
- Return car status information.*

  - [DivisionMap::const\\_iterator](#) [FirstDivision](#) () const
- Iterator of the first division in the division map.*

  - [DivisionMap::const\\_iterator](#) [LastDivision](#) () const
- Iterator of one past the last division in the division map.*

  - [StationMap::const\\_iterator](#) [FirstStation](#) () const

- Iterator of the first station in the station map.*
- StationMap::const\_iterator [LastStation](#) () const
- Iterator of one past the last station in the station map.*
- TrainMap::const\_iterator [FirstTrain](#) () const
- Iterator of the first train in the train map.*
- TrainMap::const\_iterator [LastTrain](#) () const
- Iterator of one past the last train in the train map.*
- IndustryMap::const\_iterator [FirstIndustry](#) () const
- Iterator of the first industry in the industry map.*
- IndustryMap::const\_iterator [LastIndustry](#) () const
- Iterator of one past the last industry in the industry map.*
- CarTypeMap::const\_iterator [FirstCarType](#) () const
- Iterator of the first car type in the car type map.*
- CarTypeMap::const\_iterator [LastCarType](#) () const
- Iterator of one past the last car type in the car type map.*
- OwnerMap::const\_iterator [FirstOwner](#) () const
- Iterator of the first owner in the owner map.*
- OwnerMap::const\_iterator [LastOwner](#) () const
- Iterator of one past the last owner in the owner map.*
- vector< int > [SearchForCarIndexesByNumber](#) (string number, bool subStringP) const
- Search for cars with a specificed number.*
- vector< int > [SearchForTrainPattern](#) (string trainNamePattern) const
- Search for a train by name given a glob pattern.*
- vector< int > [SearchForIndustryPattern](#) (string industryNamePattern) const
- Search for an industry by name given a glob pattern.*

## Protected Member Functions

- [System](#) ()
- The default constructor.*

## Private Member Functions

- string [trim](#) (string line) const
- Helper utility function to trim white space off the ends of a string.*
- vector< string > [split](#) (string s, char delimiter) const
- Helper utility to split a string into words.*
- bool [SkipCommentsGets](#) (istream &stream, string &buffer, const char \*message, char \*\*outmessage=NULL)
- Utility to get a line after skipping any intervening comments.*
- bool [ReadGroupLimit](#) (istream &stream, const char \*label, int &value, const char \*filename, char \*\*outmessage=NULL)
- Utility to read a group limit.*
- bool [ReadDivisions](#) (istream &stream, map< int, int, less< int > > &homemap, char \*\*outmessage=NULL)
- Read in the division map.*
- bool [ReadStations](#) (istream &stream, char \*\*outmessage=NULL)
- Read in the station map.*

- bool [ReadTrains](#) (char \*\*outmessage=NULL)  
*Read in the trains file.*
- bool [ReadIndustries](#) (char \*\*outmessage=NULL)  
*Read in the industries file.*
- bool [ReadTrainOrders](#) (char \*\*outmessage=NULL)  
*Read in the train orders file.*
- bool [ReadCarTypes](#) (char \*\*outmessage=NULL)  
*Read in the car types file.*
- bool [ReadOwners](#) (char \*\*outmessage=NULL)  
*Read in the owners file.*
- bool [StringToInt](#) (string str, int &result, const char \*message, char \*\*outmessage=NULL) const  
*Convert a string to an integer.*
- bool [StringToIntRange](#) (string str, int &result, int minv, int maxv, const char \*message, char \*\*outmessage=NULL) const  
*Convert a string to an integer and check its range.*
- bool [WriteOneCarToDisk](#) ([Car](#) \*car, ostream &stream)  
*Function to write one car to disk.*
- bool [IndustryTakesCar](#) ([Industry](#) \*Ix, [Car](#) \*Cx)  
*Check if an industry takes a certain car.*
- bool [FixedRouteMirrorCheck](#) ([Car](#) \*Cx, [Industry](#) \*Ix)  
*Check to see if a certain car can be mirrored on a fixed route at a certain industry.*
- [CarVector::iterator](#) [FindCarInCarVector](#) ([CarVector](#) &cvect, [Car](#) \*car)  
*Find a car in a car vector.*
- [IndustryMap::iterator](#) [FindIndustry](#) ([Industry](#) \*industry)  
*Find an industry in the industry map.*
- void [GetIndustryCarCounts](#) ()  
*Update industry car counts.*
- void [InternalRunOneTrain](#) ([Train](#) \*train, bool boxMove, const [FCFSupport::TrainDisplayCallback](#) \*traindisplay, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer)  
*Internal function to run a single train.*
- void [RunOneLocal](#) ([Train](#) \*train, bool boxMove, [CarVector](#) &consist, const [FCFSupport::TrainDisplayCallback](#) \*traindisplay, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer)  
*One one local train.*
- void [RunOnePassenger](#) ([Train](#) \*train, bool boxMove, const [FCFSupport::TrainDisplayCallback](#) \*traindisplay, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer)  
*One one passenger train.*
- void [RunOneManifest](#) ([Train](#) \*train, bool boxMove, [CarVector](#) &consist, const [FCFSupport::TrainDisplayCallback](#) \*traindisplay, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer)  
*Run one manifest freight train.*
- void [PrintTrainLoc](#) ([Train](#) \*train, int Px, const [FCFSupport::LogMessageCallback](#) \*Log, const [FCFSupport::TrainDisplayCallback](#) \*traindisplay)  
*Print a train's current location.*
- void [TrainLocalOriginate](#) ([Train](#) \*train, bool boxMove, int Px, [CarVector](#) &consist, bool &didAction, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer)  
*Make up a local train.*
- void [LogCarPickup](#) ([Car](#) \*car, [Train](#) \*train, bool boxMove)  
*Log a car pickup in the switch list structure.*



- void [TrainLocalDrops](#) ([Train](#) \*train, int Px, [CarVector](#) &consist, bool &didAction, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer)  
*Drop cars from a local (box move or way freight).*
- void [TrainManifestDrops](#) ([Train](#) \*train, int Px, [CarVector](#) &consist, bool &didAction, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer)  
*Drop cars from a manifest freight.*
- void [TrainDropOneCar](#) ([Car](#) \*car, [Train](#) \*train, [CarVector::iterator](#) Lx, [CarVector](#) &consist, bool &didAction, int Px, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer)  
*Drop a single car.*
- void [TrainDropAllCars](#) ([Train](#) \*train, int Px, [CarVector](#) &consist, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer)  
*Drop all cars from a train at the current stop (usually the last stop).*
- void [TrainLocalPickups](#) ([Train](#) \*train, bool boxMove, int Px, [CarVector](#) &consist, bool &didAction, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer)  
*Pick up cars for a local train (box move or way freight).*
- void [TrainManifestPickups](#) ([Train](#) \*train, bool boxMove, int Px, [CarVector](#) &consist, bool &didAction, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer)  
*Pick up cars for a manifest freight train.*
- void [TrainPickupOneCar](#) ([Car](#) \*car, [Train](#) \*train, bool boxMove, [CarVector](#) &consist, bool &didAction, int Px, [CarVector::iterator](#) Lx, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer)  
*Pick up one car.*
- bool [TrainCarPickupCheck](#) ([Car](#) \*car, [Train](#) \*train, bool boxMove, [CarVector](#) &consist, bool &didAction, int Px, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer)  
*Check to see if we can really pick up this car.*
- bool [OtherCarOkForTrain](#) ([Car](#) \*car, [Train](#) \*train)  
*Check to see if this other car can be picked up.*
- void [TrainPrintConsistSummary](#) ([Train](#) \*train, [CarVector](#) &consist, [FCFSupport::PrinterDevice](#) \*printer)  
*Print a train's consist summary.*
- void [TrainPrintFinalSummary](#) ([Train](#) \*train, [FCFSupport::PrinterDevice](#) \*printer)  
*Print a train's final summary.*
- void [TrainPrintTown](#) (const [FCFSupport::Train](#) \*train, const [FCFSupport::Station](#) \*curStation, [FCFSupport::PrinterDevice](#) \*printer)  
*Print the town a train is in.*
- void [PrintTrainOrderHeader](#) (const [FCFSupport::Train](#) \*train, [FCFSupport::PrinterDevice](#) \*printer)  
*Print a train order header.*
- void [PrintFormFeed](#) ([FCFSupport::PrinterDevice](#) \*printer) const  
*Print a form feed.*
- void [PrintSystemBanner](#) ([FCFSupport::PrinterDevice](#) \*printer) const  
*Print a system banner.*
- void [PrintDashedLine](#) ([FCFSupport::PrinterDevice](#) \*printer) const  
*Print a dashed line.*
- void [PrintDispatcher](#) (string banner, char trainType, [FCFSupport::PrinterDevice](#) \*printer) const  
*Print dispatcher report sheets.*
- const string [FormatDutyTime](#) (int dutytimeminutes) const  
*Format the on duty time in a human readable format.*
- void [PrintTrainOrders](#) (const [Train](#) \*train, [FCFSupport::PrinterDevice](#) \*printer) const  
*Print the train orders for a selected train.*
- const string [Today](#) () const

- Return today's date.*

  - const string [UpperCase](#) (const string str) const

*Convert a string to all uppercase letters.*
- void [PrintIndustryHeader](#) ([FCFSupport::PrinterDevice](#) \*printer) const

*Print the industry header.*
- void [PrintOneIndustry](#) (const [Industry](#) \*ix, int &lenInDiv, int &carsInDiv, int &carsToDiv, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer) const

*Print one industry.*
- void [PrintCarHeading](#) ([FCFSupport::PrinterDevice](#) \*printer) const

*Print the car heading.*
- void [PrintOneCarInfo](#) (const [Car](#) \*car, [FCFSupport::PrinterDevice](#) \*printer) const

*Print one car's information.*
- void [PrintCarTypesHeader](#) ([FCFSupport::PrinterDevice](#) \*printer) const

*Print the car type header.*
- void [PrintAllCarTypes](#) (bool totalsOnly, [FCFSupport::PrinterDevice](#) \*printer) const

*Print all car types.*
- void [PrintOneCarType](#) (bool totalsOnly, char carType, const [CarType](#) \*ct, int &OnLineShippersOfType, int &OfflineShippersOfType, int &OnLineReceiversOfType, int &OfflineReceiversOfType, int &allTotalMoves, int &allTotalAssigns, [FCFSupport::PrinterDevice](#) \*printer) const

*Print one car type.*
- void [PrintCarTypesSummaryHeader](#) ([FCFSupport::PrinterDevice](#) \*printer) const

*Print car type summary header.*
- void [ReportLocIndustry](#) ([IndustryMap::const\\_iterator](#) lx, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer, char \*\*outmessage=NULL)

*Print a location report for one industry.*
- void [ReportLocStation](#) ([StationMap::const\\_iterator](#) Sx, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer, char \*\*outmessage=NULL)

*Print a location report for one station.*
- void [ReportLocDivision](#) ([DivisionMap::const\\_iterator](#) Dx, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer, char \*\*outmessage=NULL)

*Print a location report for one division.*
- void [ReportLocAll](#) (bool printBench, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer, char \*\*outmessage=NULL)

*Print a location report for all locations.*
- void [PrintLocCommon](#) ([FCFSupport::PrinterDevice](#) \*printer)

*Print a header for all location reports.*
- void [PrintLocOneIndustry](#) (const [FCFSupport::Industry](#) \*ix, const [FCFSupport::Station](#) \*Sx, bool &firstOne, [FCFSupport::PrinterDevice](#) \*printer) const

*Print a location report for a single industry.*
- void [PrintOneCarLocation](#) (const [Car](#) \*car, [FCFSupport::PrinterDevice](#) \*printer) const

*Print one car location report.*
- void [PrintOneAnalysis](#) (const [Industry](#) \*ix, int &carsToDiv, const [FCFSupport::LogMessageCallback](#) \*Log, [FCFSupport::PrinterDevice](#) \*printer) const

*Print one analysis report.*
- void [PrintAnalysisHeader](#) ([FCFSupport::PrinterDevice](#) \*printer) const

*Print an analysis header.*
- bool [GlobStringMatch](#) (const string thestring, const string pattern) const

*Glob style string match function.*
- bool [GlobStringMatchHelper](#) (string::const\_iterator string\_i, string::const\_iterator string\_e, string::const\_iterator pattern\_i, string::const\_iterator pattern\_e) const

*Helper function for glob string matching.*

## Private Attributes

- [PathName systemFile](#)  
*Full pathname of the system file.*
- [string systemName](#)  
*The system name.*
- [PathName industriesFile](#)  
*Full pathname of the industries file.*
- [PathName trainsFile](#)  
*Full pathname of the trains file.*
- [PathName ordersFile](#)  
*Full pathname of the train orders file.*
- [PathName ownersFile](#)  
*Full pathname of the car owners file.*
- [PathName carTypesFile](#)  
*Full pathname of the car types file.*
- [PathName carsFile](#)  
*Full pathname of the cars file.*
- [PathName statsFile](#)  
*Full pathname of the stats file.*
- [DivisionMap divisions](#)  
*Division map.*
- [StationMap stations](#)  
*Station map.*
- [TrainMap trains](#)  
*Train map.*
- [TrainNameMap trainIndex](#)  
*Train name map.*
- [IndustryMap industries](#)  
*Industries map.*
- [char carTypesOrder \[CarType::MaxCarTypes\]](#)  
*Car type order vector.*
- [CarTypeMap carTypes](#)  
*Car type map.*
- [CarGroup \\* carGroups \[CarGroup::MaxCarGroup\]](#)  
*Car group vector.*
- [OwnerMap owners](#)  
*Car owner map.*
- [CarVector cars](#)  
*Car vector.*
- [SwitchList switchList](#)  
*Switch lists.*
- [int sessionNumber](#)  
*Current session number.*
- [int shiftNumber](#)  
*Current shift number.*
- [int totalShifts](#)

- The total number of shifts.*
- int [ranAllTrains](#)
  - The ran all trains flag.*
- int [totalPickups](#)
  - The total number of pickups.*
- int [totalLoads](#)
  - The total number of loads.*
- int [totalTons](#)
  - The total number of tons.*
- int [totalRevenueTons](#)
  - The total number of revenue tons.*
- bool [trainPrintOK](#)
  - Train print flag.*
- bool [wayFreight](#)
  - Way freight flag.*
- bool [deliver](#)
  - Deliver flag.*
- int [trainLength](#)
  - Train length.*
- int [numberCars](#)
  - The number of cars on a train.*
- int [trainTons](#)
  - The number of tons on a train.*
- int [trainLoads](#)
  - The number of loads on a train.*
- int [trainEmpties](#)
  - The number of empties on a train.*
- int [trainLongest](#)
  - The longest a train has been.*
- [Division](#) \* [curDiv](#)
  - Current division.*
- [Industry](#) \* [originYard](#)
  - Origin Yard.*
- [Industry](#) \* [trainLastLocation](#)
  - A trains last location.*
- [Industry](#) \* [carDest](#)
  - A temporary for a car's location.*
- int [statsPeriod](#)
  - The current stats period.*
- int [carsMoved](#)
  - The number of cars moved.*
- int [carsAtDest](#)
  - The number of cars at their destinations.*
- int [carsNotMoved](#)
  - The number of cars not moved.*
- int [carsMovedOnce](#)
  - The number of cars moved one time.*

- int [carsMovedTwice](#)  
*The number of cars moved two times.*
- int [carsMovedThree](#)  
*The number of cars moved three times.*
- int [carsMovedMore](#)  
*The number of cars moved more than three times.*
- int [carMovements](#)  
*The number of cars movements.*
- int [carsInTransit](#)  
*The number of cars in transit.*
- int [carsAtWorkBench](#)  
*The number of cars at the workbench.*
- int [carsAtDest\\_carsInTransit](#)  
*The number of cars at their destinations and still in transit.*
- bool [printYards](#)  
*Flag for printing yard switch lists.*
- bool [printAlpha](#)  
*Flag for printing alphabetical lists.*
- bool [printAtwice](#)  
*Flag for printing a second copy of alphabetical lists.*
- bool [printList](#)  
*Flag for printing train switch lists.*
- bool [printLtwice](#)  
*Flag for printing a second copy of train switch lists.*
- bool [printDispatch](#)  
*Flag for printing a dispatcher's report.*
- bool [printem](#)  
*Flag for printing train movements.*
- char [messageBuffer](#) [2048]  
*Message buffer, used for error messages mostly.*
- const [FCFSupport::Industry indScrapYard](#)  
*The pointer to the scrapyard.*

## Static Private Attributes

- static const string [whitespace](#)  
*String of white space characters.*

### 9.170.1 Detailed Description

This is the main Freight [Car](#) Forwarder class.

It implements all of the basic data and algorithms used in the the Freight [Car](#) Forwarder system.

This class includes code to load a model railroad "system" (divisions, stations, industries, cars, and trains) along with code to assign cars to trains, run trains, generate yard switch lists, and various reports. Basically everything you need run realistic trains on a layout.

This is my second port of Tim O'Connors Freight [Car](#) Forwarding system, originally written in QBasic for use with the North Shore Model RR Club "Chesapeake [System](#)".

**Author**

Robert Heller <heller@deepsoft.com>

**9.170.2 Member Enumeration Documentation****9.170.2.1 CarLocationType**

enum `FCFSupport::System::CarLocationType`

Types of location report.

**Enumerator**

INDUSTRY	Report by industry.
STATION	Report by station.
DIVISION	Report by division.
ALL	Report on all locations.

**9.170.2.2 CarTypeReport**

enum `FCFSupport::System::CarTypeReport`

Types of car type reports.

**Enumerator**

All	Report on all car types.
Type	Report on one type.
Summary	Report summary.

**9.170.3 Constructor & Destructor Documentation****9.170.3.1 System() [1/2]**

`FCFSupport::System::System ( ) [inline], [protected]`

The default constructor.

This is protected to prevent the creation of an uninitialized class instance. It simply makes no sense to create a system without loading a system file.

References `~System()`.

### 9.170.3.2 System() [2/2]

```
FCFSupport::System::System (
    const char * systemfile,
    int seed,
    char ** outmessage = NULL )
```

The constructor for the system.

Takes the path to a system file (typically #system.dat#) and loads the complete system. The system file contains the names of the additional files, containing the remaining system data. All of the files are presumed to exist in the same directory as the system file. All of the files are loaded and a sanity check is made to insure that the data is sane.

#### Parameters

<i>systemfile</i>	Pathname to the system file.
<i>seed</i>	Seed value for the random number generator.
<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur.

### 9.170.3.3 ~System()

```
FCFSupport::System::~~System ( )
```

The destructor frees all memory and generally cleans things up.

Referenced by `System()`.

## 9.170.4 Member Function Documentation

### 9.170.4.1 AddCar()

```
void FCFSupport::System::AddCar (
    FCFSupport::Car * newcar ) [inline]
```

Add a new car to the array of cars.

## Parameters

<i>newcar</i>	The new car.
---------------	--------------

References cars.

## 9.170.4.2 AddOwner()

```
void FCFSupport::System::AddOwner (
    const char * initials ) [inline]
```

Create a new owner given a set of initials.

## Parameters

<i>initials</i>	The new car owner's initials.
-----------------	-------------------------------

References owners.

## 9.170.4.3 CarAssignment()

```
void FCFSupport::System::CarAssignment (
    const FCFSupport::WorkInProgressCallback * WIP,
    const FCFSupport::LogMessageCallback * log,
    const FCFSupport::ShowBannerCallback * banner,
    char ** outmessage = NULL )
```

[Car](#) assignment procedure.

The is one of the main workhorse procedures. It goes through all of the cars, finding ones that are ready to be moved and determines where they could be moved to, based on a number of critiera, such as whether they are loaded or empty, whether they are in their home divisions or not, and so on.

## Parameters

<i>WIP</i>	Work in progress callback.
<i>log</i>	Log message callback.
<i>banner</i>	Show banner callback.
<i>outmessage</i>	Buffer pointer for error messages.

Referenced by Random().



#### 9.170.4.4 CarMovements()

```
int FCFSupport::System::CarMovements ( ) const [inline]
```

Return the total number of car movements.

References carMovements.

#### 9.170.4.5 CarsAtDest()

```
int FCFSupport::System::CarsAtDest ( ) const [inline]
```

Return the number of cars that are at their destinations.

References carsAtDest.

#### 9.170.4.6 CarsAtDest\_CarsInTransit()

```
int FCFSupport::System::CarsAtDest_CarsInTransit ( ) const [inline]
```

Return the number of cars at their destinations plus the number of cars in transit.

References carsAtDest\_carsInTransit.

#### 9.170.4.7 CarsAtWorkBench()

```
int FCFSupport::System::CarsAtWorkBench ( ) const [inline]
```

Return the number of cars on the RIP track (the workbench).

References carsAtWorkBench.

#### 9.170.4.8 CarsFile()

```
const char* FCFSupport::System::CarsFile ( ) const [inline]
```

Return the Cars file's full path name.

References carsFile, and FCFSupport::PathName::FullPath().

**9.170.4.9 CarsInTransit()**

```
int FCFSupport::System::CarsInTransit ( ) const [inline]
```

Return the number of cars still in transit.

References carsInTransit.

**9.170.4.10 CarsMoved()**

```
int FCFSupport::System::CarsMoved ( ) const [inline]
```

Return the number of cars moved.

References carsMoved.

**9.170.4.11 CarsMovedMore()**

```
int FCFSupport::System::CarsMovedMore ( ) const [inline]
```

Return the number of cars moved more then three times.

References carsMovedMore.

**9.170.4.12 CarsMovedOnce()**

```
int FCFSupport::System::CarsMovedOnce ( ) const [inline]
```

Return the number of cars moved once.

References carsMovedOnce.

**9.170.4.13 CarsMovedThree()**

```
int FCFSupport::System::CarsMovedThree ( ) const [inline]
```

Return the number of cars moved three times.

References carsMovedThree.

#### 9.170.4.14 CarsMovedTwice()

```
int FCFSupport::System::CarsMovedTwice ( ) const [inline]
```

Return the number of cars moved twice.

References carsMovedTwice.

#### 9.170.4.15 CarsNotMoved()

```
int FCFSupport::System::CarsNotMoved ( ) const [inline]
```

Return the number of cars not moved at all.

References carsNotMoved.

#### 9.170.4.16 CarTypesFile()

```
const char* FCFSupport::System::CarTypesFile ( ) const [inline]
```

Return the [Car](#) Types file's full path name.

References carTypesFile, and FCFSupport::PathName::FullPath().

#### 9.170.4.17 CarTypesOrder()

```
char FCFSupport::System::CarTypesOrder (
    int i ) const [inline]
```

Access a car type by index.

##### Parameters

<i>i</i>	The car type index.
----------	---------------------

References carTypesOrder, CarTypesOrderIndex(), *i*, and FCFSupport::CarType::MaxCarTypes.

**9.170.4.18 CarTypesOrderIndex()**

```
int FCFSupport::System::CarTypesOrderIndex (
    char type ) const
```

Car type order index.

Get the index of a car type.

**Parameters**

<i>type</i>	The car type to lookup.
-------------	-------------------------

Referenced by CarTypesOrder().

**9.170.4.19 DeleteAllExistingCars()**

```
void FCFSupport::System::DeleteAllExistingCars ( )
```

Delete all existing cars.

Referenced by RanAllTrains().

**9.170.4.20 FindCarInCarVector()**

```
CarVector::iterator FCFSupport::System::FindCarInCarVector (
    CarVector & cvect,
    Car * car ) [private]
```

Find a car in a car vector.

**Parameters**

<i>cvect</i>	The car vector to search.
<i>car</i>	The car to search for.

**9.170.4.21 FindDivisionByIndex()**

```
const Division* FCFSupport::System::FindDivisionByIndex (
    int i ) const [inline]
```

Find a division by its index.

Returns either a pointer to the division or NULL.

#### Parameters

<i>i</i>	The division index to look for.
----------	---------------------------------

References divisions, and FindDivisionBySymbol().

#### 9.170.4.22 FindDivisionBySymbol()

```
const Division* FCFSupport::System::FindDivisionBySymbol (
    char symbol ) const
```

Find a division by its symbol.

Returns either a pointer to the division or NULL.

#### Parameters

<i>symbol</i>	The division symbol to look for.
---------------	----------------------------------

Referenced by FindDivisionByIndex().

#### 9.170.4.23 FindDivisionIndex()

```
int FCFSupport::System::FindDivisionIndex (
    const FCFSupport::Division * division ) const
```

Find a division's index.

#### Parameters

<i>division</i>	The division to look for.
-----------------	---------------------------

#### 9.170.4.24 FindIndustry()

```
IndustryMap::iterator FCFSupport::System::FindIndustry (
    Industry * industry ) [private]
```

Find an industry in the industry map.

**Parameters**

<i>industry</i>	The industry to search for.
-----------------	-----------------------------

**9.170.4.25 FindIndustryByIndex()**

```
const FCFSupport::Industry* FCFSupport::System::FindIndustryByIndex (
    int i ) const [inline]
```

Find an industry by its index.

Returns either a pointer to the industry or NULL.

**Parameters**

<i>i</i>	The industry index to look for.
----------	---------------------------------

References FindIndustryByName(), and industries.

**9.170.4.26 FindIndustryByName()**

```
const FCFSupport::Industry* FCFSupport::System::FindIndustryByName (
    string name ) const
```

Find an industry by its name.

Returns either a pointer to the industry or NULL.

**Parameters**

<i>name</i>	Industry name to look for.
-------------	----------------------------

Referenced by FindIndustryByIndex().

**9.170.4.27 FindIndustryIndex()**

```
int FCFSupport::System::FindIndustryIndex (
    const FCFSupport::Industry * industry ) const
```

Find an industry's index.

#### Parameters

<i>industry</i>	The industry to look for.
-----------------	---------------------------

#### 9.170.4.28 FindStationByName()

```
const FCFSupport::Station* FCFSupport::System::FindStationByName (
    string name,
    string comment ) const
```

[Station](#) indexing (by name) function.

Returns NULL if the named station does not exist.

#### Parameters

<i>name</i>	<a href="#">Station</a> name to access.
-------------	---

Referenced by TheStation().

#### 9.170.4.29 FindStationIndex()

```
int FCFSupport::System::FindStationIndex (
    const FCFSupport::Station * station ) const
```

Find a station's index.

#### Parameters

<i>station</i>	The station to look for.
----------------	--------------------------

#### 9.170.4.30 FindTrainByIndex()

```
const Train* FCFSupport::System::FindTrainByIndex (
    int i ) const [inline]
```

Find a train by its index.

Returns either a pointer to the train or NULL.

**Parameters**

<i>i</i>	The train index to look for.
----------	------------------------------

References trains.

**9.170.4.31 FindTrainByName()**

```
const Train* FCFSupport::System::FindTrainByName (
    const char * name ) const [inline]
```

Find a train by its name.

Returns either a pointer to the train or NULL.

**Parameters**

<i>name</i>	<a href="#">Train</a> name to look for.
-------------	---

References trainIndex.

**9.170.4.32 FirstCarType()**

```
CarTypeMap::const_iterator FCFSupport::System::FirstCarType ( ) const [inline]
```

Iterator of the first car type in the car type map.

References carTypes.

**9.170.4.33 FirstDivision()**

```
DivisionMap::const_iterator FCFSupport::System::FirstDivision ( ) const [inline]
```

Iterator of the first division in the division map.

References divisions.



#### 9.170.4.34 FirstIndustry()

```
IndustryMap::const_iterator FCFSupport::System::FirstIndustry ( ) const [inline]
```

Iterator of the first industry in the industry map.

References industries.

#### 9.170.4.35 FirstOwner()

```
OwnerMap::const_iterator FCFSupport::System::FirstOwner ( ) const [inline]
```

Iterator of the first owner in the owner map.

References owners.

#### 9.170.4.36 FirstStation()

```
StationMap::const_iterator FCFSupport::System::FirstStation ( ) const [inline]
```

Iterator of the first station in the station map.

References stations.

#### 9.170.4.37 FirstTrain()

```
TrainMap::const_iterator FCFSupport::System::FirstTrain ( ) const [inline]
```

Iterator of the first train in the train map.

References trains.

#### 9.170.4.38 FixedRouteMirrorCheck()

```
bool FCFSupport::System::FixedRouteMirrorCheck (
    Car * Cx,
    Industry * Ix ) [private]
```

Check to see if a certain car can be mirrored on a fixed route at a certain industry.

**Parameters**

<i>Cx</i>	The car to check.
<i>Ix</i>	The industry to check.

**9.170.4.39 FormatDutyTime()**

```
const string FCFSupport::System::FormatDutyTime (
    int dutytimeminutes ) const [private]
```

Format the on duty time in a human readable format.

**Parameters**

<i>dutytimeminutes</i>	The duty time in minutes.
------------------------	---------------------------

**9.170.4.40 GetCarStatus()**

```
void FCFSupport::System::GetCarStatus (
    const Car * car,
    string & status,
    string & carTypeDescr ) const
```

Return car status information.

**Parameters**

<i>car</i>	The car to look up.
<i>status</i>	Its status (loaded or empty).
<i>carTypeDescr</i>	Its car type description.

**9.170.4.41 GetIndustryCarCounts()**

```
void FCFSupport::System::GetIndustryCarCounts ( ) [private]
```

Update industry car counts.

## 9.170.4.42 GlobStringMatch()

```
bool FCFSupport::System::GlobStringMatch (
    const string thestring,
    const string pattern ) const [private]
```

Glob style string match function.

## Parameters

<i>thestring</i>	The string to match against.
<i>pattern</i>	The glob pattern.

## 9.170.4.43 GlobStringMatchHelper()

```
bool FCFSupport::System::GlobStringMatchHelper (
    string::const_iterator string_i,
    string::const_iterator string_e,
    string::const_iterator pattern_i,
    string::const_iterator pattern_e ) const [private]
```

Helper function for glob string matching.

## Parameters

<i>string_i</i>	The current string index.
<i>string_e</i>	The end of the string.
<i>pattern↔ _i</i>	The current pattern index.
<i>pattern↔ _e</i>	The end of the pattern.

## 9.170.4.44 IndRipTrack()

```
const FCFSupport::Industry* FCFSupport::System::IndRipTrack ( ) [inline]
```

Return a pointer to the RIP track (workbench).

References industries.

**9.170.4.45 IndRipTrackConst()**

```
const FCFSupport::Industry* FCFSupport::System::IndRipTrackConst ( ) const [inline]
```

Const version of the pointer to the RIP track (workbench).

References industries, and RestartLoop().

**9.170.4.46 IndScrapYard()**

```
const FCFSupport::Industry* FCFSupport::System::IndScrapYard ( ) const [inline]
```

Return a pointer to the scrap yard.

References indScrapYard.

**9.170.4.47 IndustriesFile()**

```
const char* FCFSupport::System::IndustriesFile ( ) const [inline]
```

Return the industry file's full path name.

References FCFSupport::PathName::FullPath(), and industriesFile.

**9.170.4.48 IndustryIndex()**

```
int FCFSupport::System::IndustryIndex (
    const FCFSupport::Industry * indus ) const
```

Return an industry's index.

**Parameters**

<i>indus</i>	The industry to lookup.
--------------	-------------------------

Referenced by StatsPeriod().

#### 9.170.4.49 IndustryTakesCar()

```
bool FCFSupport::System::IndustryTakesCar (
    Industry * Ix,
    Car * Cx ) [private]
```

Check if an industry takes a certain car.

##### Parameters

<i>Ix</i>	The industry to check.
<i>Cx</i>	The car to check.

#### 9.170.4.50 InternalRunOneTrain()

```
void FCFSupport::System::InternalRunOneTrain (
    Train * train,
    bool boxMove,
    const FCFSupport::TrainDisplayCallback * traindisplay,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer ) [private]
```

Internal function to run a single train.

##### Parameters

<i>train</i>	The train to run.
<i>boxMove</i>	Is this a box move?
<i>traindisplay</i>	Train display callback.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.

#### 9.170.4.51 LastCarType()

```
CarTypeMap::const_iterator FCFSupport::System::LastCarType ( ) const [inline]
```

Iterator of one past the last car type in the car type map.

References carTypes.

**9.170.4.52 LastDivision()**

```
DivisionMap::const_iterator FCFSupport::System::LastDivision ( ) const [inline]
```

Iterator of one past the last division in the division map.

References divisions.

**9.170.4.53 LastIndustry()**

```
IndustryMap::const_iterator FCFSupport::System::LastIndustry ( ) const [inline]
```

Iterator of one past the last industry in the industry map.

References industries.

**9.170.4.54 LastOwner()**

```
OwnerMap::const_iterator FCFSupport::System::LastOwner ( ) const [inline]
```

Iterator of one past the last owner in the owner map.

References owners, SearchForCarIndexesByNumber(), SearchForIndustryPattern(), and SearchForTrainPattern().

**9.170.4.55 LastStation()**

```
StationMap::const_iterator FCFSupport::System::LastStation ( ) const [inline]
```

Iterator of one past the last station in the station map.

References stations.

**9.170.4.56 LastTrain()**

```
TrainMap::const_iterator FCFSupport::System::LastTrain ( ) const [inline]
```

Iterator of one past the last train in the train map.

References trains.

**9.170.4.57 LoadCarFile()**

```
bool FCFSupport::System::LoadCarFile (
    char ** outmessage = NULL )
```

(Re-)Load the car file.

## Parameters

<i>outmessage</i>	Buffer pointer for error messages.
-------------------	------------------------------------

Referenced by `RanAllTrains()`, and `ReLoadCarFile()`.

**9.170.4.58 LoadStatsFile()**

```
bool FCFSupport::System::LoadStatsFile (
    char ** outmessage = NULL )
```

Load the stats file.

## Parameters

<i>outmessage</i>	Buffer pointer for error messages.
-------------------	------------------------------------

Referenced by `RanAllTrains()`, and `ReLoadCarFile()`.

**9.170.4.59 LogCarPickup()**

```
void FCFSupport::System::LogCarPickup (
    Car * car,
    Train * train,
    bool boxMove ) [private]
```

Log a car pickup in the switch list structure.

## Parameters

<i>car</i>	The car picked up.
<i>train</i>	The train that picked it up.
<i>boxMove</i>	Is this a box move?

**9.170.4.60 NextShift()**

```
int FCFSupport::System::NextShift ( ) [inline]
```

Increment the shift number.

References sessionNumber, shiftNumber, and totalShifts.

#### 9.170.4.61 NumberOfCars()

```
int FCFSupport::System::NumberOfCars ( ) const [inline]
```

References cars.

#### 9.170.4.62 NumberOfDivisions()

```
int FCFSupport::System::NumberOfDivisions ( ) const [inline]
```

return the number of divisions loaded.

References divisions.

#### 9.170.4.63 NumberOfIndustries()

```
int FCFSupport::System::NumberOfIndustries ( ) const [inline]
```

Return the number of industries loaded.

References industries.

#### 9.170.4.64 NumberOfStations()

```
int FCFSupport::System::NumberOfStations ( ) const [inline]
```

The number of stations loaded.

References stations.



#### 9.170.4.65 NumberOfTrains()

```
int FCFSupport::System::NumberOfTrains ( ) const [inline]
```

The number of trains loaded.

References trains.

#### 9.170.4.66 OrdersFile()

```
const char* FCFSupport::System::OrdersFile ( ) const [inline]
```

Return the train orders file's full path name.

References FCFSupport::PathName::FullPath(), and ordersFile.

#### 9.170.4.67 OtherCarOkForTrain()

```
bool FCFSupport::System::OtherCarOkForTrain (
    Car * car,
    Train * train ) [private]
```

Check to see if this other car can be picked up.

##### Parameters

<i>car</i>	The car to check.
<i>train</i>	The train to check.

#### 9.170.4.68 OwnersFile()

```
const char* FCFSupport::System::OwnersFile ( ) const [inline]
```

Return the Owners file's full path name.

References FCFSupport::PathName::FullPath(), and ownersFile.

#### 9.170.4.69 PrintAllCarTypes()

```
void FCFSupport::System::PrintAllCarTypes (
    bool totalsOnly,
    FCFSupport::PrinterDevice * printer ) const [private]
```

Print all car types.

##### Parameters

<i>totalsOnly</i>	Print only the totals?
<i>printer</i>	Printer device.

#### 9.170.4.70 PrintAllLists()

```
void FCFSupport::System::PrintAllLists (
    const FCFSupport::LogMessageCallback * Log,
    const FCFSupport::ShowBannerCallback * banner,
    FCFSupport::PrinterDevice * printer )
```

Print all of the various yard and switch lists.

##### Parameters

<i>Log</i>	Log message callback.
<i>banner</i>	Show banner callback.
<i>printer</i>	Printer device.

Referenced by Random().

#### 9.170.4.71 PrintAlpha()

```
bool FCFSupport::System::PrintAlpha ( ) const [inline]
```

Print the alphabetical listing flag.

References printAlpha.

#### 9.170.4.72 PrintAnalysisHeader()

```
void FCFSupport::System::PrintAnalysisHeader (
    FCFSupport::PrinterDevice * printer ) const [private]
```

Print an analysis header.

## Parameters

<i>printer</i>	Printer device.
----------------	-----------------

## 9.170.4.73 PrintAttwice()

```
bool FCFSupport::System::PrintAttwice ( ) const [inline]
```

Print second copy of the alphabetical listing flag.

References printAttwice.

## 9.170.4.74 PrintCarHeading()

```
void FCFSupport::System::PrintCarHeading (
    FCFSupport::PrinterDevice * printer ) const [private]
```

Print the car heading.

## Parameters

<i>printer</i>	Printer device.
----------------	-----------------

## 9.170.4.75 PrintCarTypesHeader()

```
void FCFSupport::System::PrintCarTypesHeader (
    FCFSupport::PrinterDevice * printer ) const [private]
```

Print the car type header.

## Parameters

<i>printer</i>	Printer device.
----------------	-----------------

**9.170.4.76 PrintCarTypesSummaryHeader()**

```
void FCFSupport::System::PrintCarTypesSummaryHeader (
    FCFSupport::PrinterDevice * printer ) const [private]
```

Print car type summary header.

**Parameters**

<i>printer</i>	Printer device.
----------------	-----------------

**9.170.4.77 PrintDashedLine()**

```
void FCFSupport::System::PrintDashedLine (
    FCFSupport::PrinterDevice * printer ) const [private]
```

Print a dashed line.

**Parameters**

<i>printer</i>	Printer device.
----------------	-----------------

**9.170.4.78 PrintDispatch()**

```
bool FCFSupport::System::PrintDispatch ( ) const [inline]
```

Print dispatcher report sheet.

References printDispatch.

**9.170.4.79 PrintDispatcher()**

```
void FCFSupport::System::PrintDispatcher (
    string banner,
    char trainType,
    FCFSupport::PrinterDevice * printer ) const [private]
```

Print dispatcher report sheets.

## Parameters

<i>banner</i>	<a href="#">System</a> banner.
<i>trainType</i>	Type of train.
<i>printer</i>	Printer device.

## 9.170.4.80 Printem()

```
bool FCFSupport::System::Printem ( ) const [inline]
```

Print train enroute switch list.

References printem.

## 9.170.4.81 PrintFormFeed()

```
void FCFSupport::System::PrintFormFeed (
    FCFSupport::PrinterDevice * printer ) const [private]
```

Print a form feed.

## Parameters

<i>printer</i>	Printer device.
----------------	-----------------

## 9.170.4.82 PrintIndustryHeader()

```
void FCFSupport::System::PrintIndustryHeader (
    FCFSupport::PrinterDevice * printer ) const [private]
```

Print the industry header.

## Parameters

<i>printer</i>	Printer device.
----------------	-----------------

**9.170.4.83 PrintList()**

```
bool FCFSupport::System::PrintList ( ) const [inline]
```

Print the switch list order flag.

References printList.

**9.170.4.84 PrintLocCommon()**

```
void FCFSupport::System::PrintLocCommon (
    FCFSupport::PrinterDevice * printer ) [private]
```

Print a header for all location reports.

**Parameters**

<i>printer</i>	Printer device.
----------------	-----------------

**9.170.4.85 PrintLocOneIndustry()**

```
void FCFSupport::System::PrintLocOneIndustry (
    const FCFSupport::Industry * Ix,
    const FCFSupport::Station * Sx,
    bool & firstOne,
    FCFSupport::PrinterDevice * printer ) const [private]
```

Print a location report for a single industry.

**Parameters**

<i>Ix</i>	The industry to print a report for.
<i>Sx</i>	The station to print a report for.
<i>firstOne</i>	Is this the first one?
<i>printer</i>	Printer device.

**9.170.4.86 PrintLtwise()**

```
bool FCFSupport::System::PrintLtwise ( ) const [inline]
```

Print a second copy of the switch list order flag.

References `printLt` twice.

#### 9.170.4.87 `PrintOneAnalysis()`

```
void FCFSupport::System::PrintOneAnalysis (
    const Industry * Ix,
    int & carsToDiv,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer ) const [private]
```

Print one analysis report.

##### Parameters

<i>Ix</i>	The industry.
<i>carsToDiv</i>	Updated cars headed for the current division.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.

#### 9.170.4.88 `PrintOneCarInfo()`

```
void FCFSupport::System::PrintOneCarInfo (
    const Car * car,
    FCFSupport::PrinterDevice * printer ) const [private]
```

Print one car's information.

##### Parameters

<i>car</i>	The car whose information to print.
<i>printer</i>	Printer device.

#### 9.170.4.89 `PrintOneCarLocation()`

```
void FCFSupport::System::PrintOneCarLocation (
    const Car * car,
    FCFSupport::PrinterDevice * printer ) const [private]
```

Print one car location report.

## Parameters

<i>car</i>	The car to print location information for.
<i>printer</i>	Printer device.

## 9.170.4.90 PrintOneCarType()

```
void FCFSupport::System::PrintOneCarType (
    bool totalsOnly,
    char carType,
    const CarType * ct,
    int & OnLineShippersOfType,
    int & OffLineShippersOfType,
    int & OnLineReceiversOfType,
    int & OffLineReceiversOfType,
    int & allTotalMoves,
    int & allTotalAssigns,
    FCFSupport::PrinterDevice * printer ) const [private]
```

Print one car type.

## Parameters

<i>totalsOnly</i>	Print only the totals?
<i>carType</i>	The car type character.
<i>ct</i>	The car type object.
<i>OnLineShippersOfType</i>	Updated online shippers of this car type.
<i>OffLineShippersOfType</i>	Updated offline shippers of this car type.
<i>OnLineReceiversOfType</i>	Updated online receivers of this car type.
<i>OffLineReceiversOfType</i>	Updated offline receivers of this car type.
<i>allTotalMoves</i>	Update total moves.
<i>allTotalAssigns</i>	Updated total assignments.
<i>printer</i>	Printer device.

## 9.170.4.91 PrintOneIndustry()

```
void FCFSupport::System::PrintOneIndustry (
    const Industry * ix,
    int & lenInDiv,
    int & carsInDiv,
    int & carsToDiv,
```



```
const FCFSupport::LogMessageCallback * Log,
FCFSupport::PrinterDevice * printer ) const [private]
```

Print one industry.

#### Parameters

<i>ix</i>	The industry.
<i>lenInDiv</i>	The updated division length.
<i>carsInDiv</i>	The updated cars in division count.
<i>carsToDiv</i>	The updates cars headed to division count.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.

#### 9.170.4.92 PrintSystemBanner()

```
void FCFSupport::System::PrintSystemBanner (
    FCFSupport::PrinterDevice * printer ) const [private]
```

Print a system banner.

#### Parameters

<i>printer</i>	Printer device.
----------------	-----------------

#### 9.170.4.93 PrintTrainLoc()

```
void FCFSupport::System::PrintTrainLoc (
    Train * train,
    int Px,
    const FCFSupport::LogMessageCallback * Log,
    const FCFSupport::TrainDisplayCallback * traindisplay ) [private]
```

Print a train's current location.

#### Parameters

<i>train</i>	The train to print.
<i>Px</i>	The stop number that train is at.
<i>Log</i>	Log message callback.
<i>traindisplay</i>	Train display callback.

#### 9.170.4.94 PrintTrainOrderHeader()

```
void FCFSupport::System::PrintTrainOrderHeader (
    const FCFSupport::Train * train,
    FCFSupport::PrinterDevice * printer ) [private]
```

Print a train order header.

##### Parameters

<i>train</i>	The train to print a train order header for.
<i>printer</i>	Printer device.

#### 9.170.4.95 PrintTrainOrders()

```
void FCFSupport::System::PrintTrainOrders (
    const Train * train,
    FCFSupport::PrinterDevice * printer ) const [private]
```

Print the train orders for a selected train.

##### Parameters

<i>train</i>	The train to print trains orders for.
<i>printer</i>	Printer device.

#### 9.170.4.96 PrintYards()

```
bool FCFSupport::System::PrintYards ( ) const [inline]
```

Print yard lists flag.

References printYards.

#### 9.170.4.97 RanAllTrains()

```
int FCFSupport::System::RanAllTrains ( ) const [inline]
```

Ran all trains?

References DeleteAllExistingCars(), LoadCarFile(), LoadStatsFile(), ranAllTrains, and SaveCars().

#### 9.170.4.98 Random()

```
double FCFSupport::System::Random ( ) [inline]
```

Return a random number between 0.0 and 1.0.

References CarAssignment(), PrintAllLists(), RunAllTrains(), RunBoxMoves(), RunOneTrain(), ShowCarMovements(), ShowCarsInDivision(), ShowCarsNotMoved(), ShowTrainCars(), ShowTrainTotals(), and ShowUnassignedCars().

#### 9.170.4.99 Randomize()

```
void FCFSupport::System::Randomize (
    int seed ) [inline]
```

Set the random seed.

##### Parameters

<i>seed</i>	Seed value.
-------------	-------------

#### 9.170.4.100 ReadCarTypes()

```
bool FCFSupport::System::ReadCarTypes (
    char ** outmessage = NULL ) [private]
```

Read in the car types file.

##### Parameters

<i>outmessage</i>	Buffer pointer for error messages.
-------------------	------------------------------------

**9.170.4.101 ReadDivisions()**

```
bool FCFSupport::System::ReadDivisions (
    istream & stream,
    map< int, int, less< int > > & homemap,
    char ** outmessage = NULL ) [private]
```

Read in the division map.

**Parameters**

<i>stream</i>	The input stream to read from.
<i>homemap</i>	The map of home yards.
<i>outmessage</i>	Buffer pointer for error messages.

**9.170.4.102 ReadGroupLimit()**

```
bool FCFSupport::System::ReadGroupLimit (
    istream & stream,
    const char * label,
    int & value,
    const char * filename,
    char ** outmessage = NULL ) [private]
```

Utility to read a group limit.

**Parameters**

<i>stream</i>	The input stream to read from.
<i>label</i>	The label for the group limit.
<i>value</i>	The limit value read.
<i>filename</i>	The filename being read from.
<i>outmessage</i>	Buffer pointer for error messages.

**9.170.4.103 ReadIndustries()**

```
bool FCFSupport::System::ReadIndustries (
    char ** outmessage = NULL ) [private]
```

Read in the industries file.

## Parameters

<i>outmessage</i>	Buffer pointer for error messages.
-------------------	------------------------------------

## 9.170.4.104 ReadOwners()

```
bool FCFSupport::System::ReadOwners (
    char ** outmessage = NULL ) [private]
```

Read in the owners file.

## Parameters

<i>outmessage</i>	Buffer pointer for error messages.
-------------------	------------------------------------

## 9.170.4.105 ReadStations()

```
bool FCFSupport::System::ReadStations (
    istream & stream,
    char ** outmessage = NULL ) [private]
```

Read in the station map.

## Parameters

<i>stream</i>	The input stream to read from.
<i>outmessage</i>	Buffer pointer for error messages.

## 9.170.4.106 ReadTrainOrders()

```
bool FCFSupport::System::ReadTrainOrders (
    char ** outmessage = NULL ) [private]
```

Read in the train orders file.

## Parameters

<i>outmessage</i>	Buffer pointer for error messages.
-------------------	------------------------------------

**9.170.4.107 ReadTrains()**

```
bool FCFSupport::System::ReadTrains (
    char ** outmessage = NULL ) [private]
```

Read in the trains file.

**Parameters**

<i>outmessage</i>	Buffer pointer for error messages.
-------------------	------------------------------------

**9.170.4.108 ReLoadCarFile()**

```
void FCFSupport::System::ReLoadCarFile (
    char ** outmessage ) [inline]
```

Reload car file.

**Parameters**

<i>outmessage</i>	Buffer pointer for error messages.
-------------------	------------------------------------

References LoadCarFile(), LoadStatsFile(), ReportCars(), ReportCarsNotMoved(), ReportIndustries(), ReportTrains(), ResetIndustryStats(), and RestartLoop().

**9.170.4.109 ReportAnalysis()**

```
void FCFSupport::System::ReportAnalysis (
    const FCFSupport::WorkInProgressCallback * WIP,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer,
    char ** outmessage = NULL ) const
```

Industry analysis report.

**Parameters**

<i>WIP</i>	Work in progress callback.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.
<i>outmessage</i>	Buffer pointer for error messages.

#### 9.170.4.110 ReportCarLocations()

```
void FCFSupport::System::ReportCarLocations (
    CarLocationType cltype,
    int index,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer,
    char ** outmessage = NULL )
```

Car location report.

##### Parameters

<i>cltype</i>	Type of report.
<i>index</i>	Index of thing to report by (industry, station, or division).
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.
<i>outmessage</i>	Buffer pointer for error messages.

#### 9.170.4.111 ReportCarOwners()

```
void FCFSupport::System::ReportCarOwners (
    string ownerInitials,
    const FCFSupport::WorkInProgressCallback * WIP,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer,
    char ** outmessage = NULL ) const
```

Report on a specified car owner.

##### Parameters

<i>ownerInitials</i>	Car owner's initials to report on.
<i>WIP</i>	Work in progress callback.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.
<i>outmessage</i>	Buffer pointer for error messages.

#### 9.170.4.112 ReportCars()

```
void FCFSupport::System::ReportCars (
    const FCFSupport::WorkInProgressCallback * WIP,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer,
    char ** outmessage = NULL ) const
```

Report on all cars.

##### Parameters

<i>WIP</i>	Work in progress callback.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.
<i>outmessage</i>	Buffer pointer for error messages.

Referenced by ReLoadCarFile().

#### 9.170.4.113 ReportCarsNotMoved()

```
void FCFSupport::System::ReportCarsNotMoved (
    const FCFSupport::WorkInProgressCallback * WIP,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer,
    char ** outmessage = NULL ) const
```

Report on cars not moved.

##### Parameters

<i>WIP</i>	Work in progress callback.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.
<i>outmessage</i>	Buffer pointer for error messages.

Referenced by ReLoadCarFile().

#### 9.170.4.114 ReportCarTypes()

```
void FCFSupport::System::ReportCarTypes (
    CarTypeReport rtype,
```



```
char carType,  
FCFSupport::PrinterDevice * printer,  
char ** outmessage = NULL ) const
```

Report on car types.

#### Parameters

<i>rtype</i>	Type of report to produce.
<i>carType</i>	Car type to report on (only used when the report type is for a single type).
<i>printer</i>	Printer device.
<i>outmessage</i>	Buffer pointer for error messages.

#### 9.170.4.115 ReportIndustries()

```
void FCFSupport::System::ReportIndustries (  
    const FCFSupport::WorkInProgressCallback * WIP,  
    const FCFSupport::LogMessageCallback * Log,  
    FCFSupport::PrinterDevice * printer,  
    char ** outmessage = NULL ) const
```

Report on all industries.

#### Parameters

<i>WIP</i>	Work in progress callback.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.
<i>outmessage</i>	Buffer pointer for error messages.

Referenced by ReLoadCarFile().

#### 9.170.4.116 ReportLocAll()

```
void FCFSupport::System::ReportLocAll (  
    bool printBench,  
    const FCFSupport::LogMessageCallback * Log,  
    FCFSupport::PrinterDevice * printer,  
    char ** outmessage = NULL ) [private]
```

Print a location report for all locations.

## Parameters

<i>printBench</i>	Print cars at the workbench?
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.
<i>outmessage</i>	Buffer pointer for error messages.

## 9.170.4.117 ReportLocDivision()

```
void FCFSupport::System::ReportLocDivision (
    DivisionMap::const_iterator Dx,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer,
    char ** outmessage = NULL ) [private]
```

Print a location report for one division.

## Parameters

<i>Dx</i>	The division's index.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.
<i>outmessage</i>	Buffer pointer for error messages.

## 9.170.4.118 ReportLocIndustry()

```
void FCFSupport::System::ReportLocIndustry (
    IndustryMap::const_iterator Ix,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer,
    char ** outmessage = NULL ) [private]
```

Print a location report for one industry.

## Parameters

<i>Ix</i>	The industry's index.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.
<i>outmessage</i>	Buffer pointer for error messages.

#### 9.170.4.119 ReportLocStation()

```
void FCFSupport::System::ReportLocStation (
    StationMap::const_iterator Sx,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer,
    char ** outmessage = NULL ) [private]
```

Print a location report for one station.

##### Parameters

<i>Sx</i>	The station's index.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.
<i>outmessage</i>	Buffer pointer for error messages.

#### 9.170.4.120 ReportTrains()

```
void FCFSupport::System::ReportTrains (
    const FCFSupport::WorkInProgressCallback * WIP,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer,
    char ** outmessage = NULL ) const
```

Report on all trains.

##### Parameters

<i>WIP</i>	Work in progress callback.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.
<i>outmessage</i>	Buffer pointer for error messages.

Referenced by ReLoadCarFile().

#### 9.170.4.121 ResetIndustryStats()

```
void FCFSupport::System::ResetIndustryStats ( )
```

Reset industry statistics.

Referenced by ReLoadCarFile().

#### 9.170.4.122 RestartLoop()

```
void FCFSupport::System::RestartLoop ( )
```

Reset loop variables.

Referenced by IndRipTrackConst(), and ReLoadCarFile().

#### 9.170.4.123 RunAllTrains()

```
void FCFSupport::System::RunAllTrains (
    const FCFSupport::WorkInProgressCallback * WIP,
    const FCFSupport::LogMessageCallback * Log,
    const FCFSupport::ShowBannerCallback * banner,
    FCFSupport::PrinterDevice * printer,
    const FCFSupport::TrainDisplayCallback * traindisplay )
```

Run all trains procedure.

The is another workhorse procedure. This procedure runs the initial box moves, then the way freights and manifest trains. It is necessary to run the box moves again after running this procedure, unless additional sections of the way freights or manifest trains need to be run first.

##### Parameters

<i>WIP</i>	Work in progress callback.
<i>Log</i>	Log message callback.
<i>banner</i>	Show banner callback.
<i>printer</i>	Printer device.
<i>traindisplay</i>	<a href="#">Train</a> display callback.

Referenced by Random().

#### 9.170.4.124 RunBoxMoves()

```
void FCFSupport::System::RunBoxMoves (
    const FCFSupport::WorkInProgressCallback * WIP,
```

```

const FCFSupport::LogMessageCallback * Log,
const FCFSupport::ShowBannerCallback * banner,
FCFSupport::PrinterDevice * printer,
const FCFSupport::TrainDisplayCallback * traindisplay )

```

Run all boxmove trains.

The is another workhorse procedure. This procedure runs all of the box moves.

#### Parameters

<i>WIP</i>	Work in progress callback.
<i>Log</i>	Log message callback.
<i>banner</i>	Show banner callback.
<i>printer</i>	Printer device.
<i>traindisplay</i>	<a href="#">Train</a> display callback.

Referenced by Random().

#### 9.170.4.125 RunOneLocal()

```

void FCFSupport::System::RunOneLocal (
    Train * train,
    bool boxMove,
    CarVector & consist,
    const FCFSupport::TrainDisplayCallback * traindisplay,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer ) [private]

```

One one local train.

#### Parameters

<i>train</i>	The train to run.
<i>boxMove</i>	Is this a box move?
<i>consist</i>	The train's consist.
<i>traindisplay</i>	<a href="#">Train</a> display callback.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.

#### 9.170.4.126 RunOneManifest()

```

void FCFSupport::System::RunOneManifest (
    Train * train,

```

```

bool boxMove,
CarVector & consist,
const FCFSupport::TrainDisplayCallback * traindisplay,
const FCFSupport::LogMessageCallback * Log,
FCFSupport::PrinterDevice * printer ) [private]

```

Run one manifest freight train.

#### Parameters

<i>train</i>	The train to run.
<i>boxMove</i>	Is this a box move?
<i>consist</i>	The train's consist.
<i>traindisplay</i>	<a href="#">Train</a> display callback.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.

#### 9.170.4.127 RunOnePassenger()

```

void FCFSupport::System::RunOnePassenger (
    Train * train,
    bool boxMove,
    const FCFSupport::TrainDisplayCallback * traindisplay,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer ) [private]

```

One one passenger train.

#### Parameters

<i>train</i>	The train to run.
<i>boxMove</i>	Is this a box move?
<i>traindisplay</i>	<a href="#">Train</a> display callback.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.

#### 9.170.4.128 RunOneTrain()

```

void FCFSupport::System::RunOneTrain (
    Train * train,
    bool boxMove,
    const FCFSupport::TrainDisplayCallback * traindisplay,

```

```
const FCFSupport::LogMessageCallback * Log,
FCFSupport::PrinterDevice * printer )
```

Run one single train.

#### Parameters

<i>train</i>	The train to run.
<i>boxMove</i>	Is this a box move?
<i>traindisplay</i>	<a href="#">Train</a> display callback.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.

Referenced by Random().

#### 9.170.4.129 SaveCars()

```
bool FCFSupport::System::SaveCars (
    char ** outmessage = NULL )
```

Save cars (and stats).

#### Parameters

<i>outmessage</i>	Buffer pointer for error messages.
-------------------	------------------------------------

Referenced by RanAllTrains().

#### 9.170.4.130 SearchForCarIndexesByNumber()

```
vector<int> FCFSupport::System::SearchForCarIndexesByNumber (
    string number,
    bool subStringP ) const
```

Search for cars with a specified number.

#### Parameters

<i>number</i>	The number string to look for.
<i>subStringP</i>	Match the whole number or only the last few digits.

Referenced by LastOwner().

#### 9.170.4.131 SearchForIndustryPattern()

```
vector<int> FCFSupport::System::SearchForIndustryPattern (
    string industryNamePattern ) const
```

Search for an industry by name given a glob pattern.

##### Parameters

<i>industryNamePattern</i>	The name pattern.
----------------------------	-------------------

Referenced by LastOwner().

#### 9.170.4.132 SearchForTrainPattern()

```
vector<int> FCFSupport::System::SearchForTrainPattern (
    string trainNamePattern ) const
```

Search for a train by name given a glob pattern.

##### Parameters

<i>trainNamePattern</i>	The name pattern.
-------------------------	-------------------

Referenced by LastOwner().

#### 9.170.4.133 SessionNumber()

```
int FCFSupport::System::SessionNumber ( ) const [inline]
```

Return the session number.

References sessionNumber.



#### 9.170.4.134 SetPrintAlpha()

```
void FCFSupport::System::SetPrintAlpha (
    bool flag ) [inline]
```

Set the print alphabetical listing flag.

**Parameters**

<i>flag</i>	New value to set the flag to.
-------------	-------------------------------

References printAlpha.

**9.170.4.135 SetPrintAtwice()**

```
void FCFSupport::System::SetPrintAtwice (  
    bool flag ) [inline]
```

Set the print second copy of the alphabetical listing flag.

**Parameters**

<i>flag</i>	New value to set the flag to.
-------------	-------------------------------

References printAtwice.

**9.170.4.136 SetPrintDispatch()**

```
void FCFSupport::System::SetPrintDispatch (  
    bool flag ) [inline]
```

Set the print dispatcher report sheet.

**Parameters**

<i>flag</i>	New value to set the flag to.
-------------	-------------------------------

References printDispatch.

**9.170.4.137 SetPrintem()**

```
void FCFSupport::System::SetPrintem (  
    bool flag ) [inline]
```

Set the print train enroute switch list.

## Parameters

<i>flag</i>	New value to set the flag to.
-------------	-------------------------------

References printem.

**9.170.4.138 SetPrintList()**

```
void FCFSupport::System::SetPrintList (  
    bool flag ) [inline]
```

Set the print switch list order flag.

## Parameters

<i>flag</i>	New value to set the flag to.
-------------	-------------------------------

References printList.

**9.170.4.139 SetPrintLtwice()**

```
void FCFSupport::System::SetPrintLtwice (  
    bool flag ) [inline]
```

Set the print a second copy of the switch list order flag.

## Parameters

<i>flag</i>	New value to set the flag to.
-------------	-------------------------------

References printLtwice.

**9.170.4.140 SetPrintYards()**

```
void FCFSupport::System::SetPrintYards (  
    bool flag ) [inline]
```

Set the print yard lists flag.

**Parameters**

<i>flag</i>	New value to set the flag to.
-------------	-------------------------------

References printYards.

**9.170.4.141 ShiftNumber()**

```
int FCFSupport::System::ShiftNumber ( ) const [inline]
```

Return the shift number.

References shiftNumber.

**9.170.4.142 ShowCarMovements()**

```
void FCFSupport::System::ShowCarMovements (
    bool showAll,
    const FCFSupport::Industry * Ix,
    const FCFSupport::Train * Tx,
    const FCFSupport::LogMessageCallback * Log,
    const FCFSupport::ShowBannerCallback * banner ) const
```

Show all car movements.

**Parameters**

<i>showAll</i>	Show all movements?
<i>Ix</i>	Show movements by industry.
<i>Tx</i>	Show movements by train.
<i>Log</i>	Log message callback.
<i>banner</i>	Show banner callback.

Referenced by Random().

**9.170.4.143 ShowCarsInDivision()**

```
void FCFSupport::System::ShowCarsInDivision (
    const Division * division,
```

```
const FCFSupport::LogMessageCallback * Log,  
const FCFSupport::ShowBannerCallback * banner ) const
```

Show cars in a specified division.

#### Parameters

<i>division</i>	The specific division.
<i>Log</i>	Log message callback.
<i>banner</i>	Show banner callback.

Referenced by Random().

#### 9.170.4.144 ShowCarsNotMoved()

```
void FCFSupport::System::ShowCarsNotMoved (  
    const FCFSupport::LogMessageCallback * Log,  
    const FCFSupport::ShowBannerCallback * banner ) const
```

Display cars not moved.

#### Parameters

<i>Log</i>	Log message callback.
<i>banner</i>	Show banner callback.

Referenced by Random().

#### 9.170.4.145 ShowTrainCars()

```
void FCFSupport::System::ShowTrainCars (  
    const Train * Tx,  
    const FCFSupport::LogMessageCallback * Log,  
    const FCFSupport::ShowBannerCallback * banner ) const
```

Show cars moved by a specific train.

#### Parameters

<i>Tx</i>	The specific train.
<i>Log</i>	Log message callback.
<i>banner</i>	Show banner callback.

Referenced by Random().

#### 9.170.4.146 ShowTrainTotals()

```
void FCFSupport::System::ShowTrainTotals (
    const FCFSupport::LogMessageCallback * Log,
    const FCFSupport::ShowBannerCallback * banner ) const
```

Show train totals.

##### Parameters

<i>Log</i>	Log message callback.
<i>banner</i>	Show banner callback.

Referenced by Random().

#### 9.170.4.147 ShowUnassignedCars()

```
void FCFSupport::System::ShowUnassignedCars (
    const FCFSupport::LogMessageCallback * Log,
    const FCFSupport::ShowBannerCallback * banner ) const
```

Show unassigned cars.

##### Parameters

<i>Log</i>	Log message callback.
<i>banner</i>	Show banner callback.

Referenced by Random().

#### 9.170.4.148 SkipCommentsGets()

```
bool FCFSupport::System::SkipCommentsGets (
    istream & stream,
    string & buffer,
    const char * message,
    char ** outmessage = NULL ) [private]
```

Utility to get a line after skipping any intervening comments.

## Parameters

<i>stream</i>	The input stream to read from.
<i>buffer</i>	The result buffer.
<i>message</i>	Error message to use if an error occurs.
<i>outmessage</i>	Buffer pointer for error messages.

## 9.170.4.149 split()

```
vector<string> FCFSupport::System::split (
    string s,
    char delimiter ) const [private]
```

Helper utility to split a string into words.

## Parameters

<i>s</i>	The string to split.
<i>delimiter</i>	The delimiter character to split the string on.

## 9.170.4.150 StatsFile()

```
const char* FCFSupport::System::StatsFile ( ) const [inline]
```

Return the Statistics file's full path name.

References FCFSupport::PathName::FullPath(), and statsFile.

## 9.170.4.151 StatsPeriod()

```
int FCFSupport::System::StatsPeriod ( ) const [inline]
```

Return the current stats period.

References IndustryIndex(), statsPeriod, and TrainIndex().

#### 9.170.4.152 StringToInt()

```
bool FCFSupport::System::StringToInt (
    string str,
    int & result,
    const char * message,
    char ** outmessage = NULL ) const [private]
```

Convert a string to an integer.

##### Parameters

<i>str</i>	The string to convert.
<i>result</i>	The converted integer result buffer.
<i>message</i>	The message to use in case there is an error.
<i>outmessage</i>	Buffer pointer for error messages.

#### 9.170.4.153 StringToIntRange()

```
bool FCFSupport::System::StringToIntRange (
    string str,
    int & result,
    int minv,
    int maxv,
    const char * message,
    char ** outmessage = NULL ) const [private]
```

Convert a string to an integer and check its range.

##### Parameters

<i>str</i>	The string to convert.
<i>result</i>	The converted integer result buffer.
<i>minv</i>	The permitted minimum value.
<i>maxv</i>	The permitted maximum value.
<i>message</i>	The message to use in case there is an error.
<i>outmessage</i>	Buffer pointer for error messages.

#### 9.170.4.154 SystemFile()

```
const char* FCFSupport::System::SystemFile ( ) const [inline]
```



Return the system file's full path name.

References FCFSupport::PathName::FullPath(), and systemFile.

#### 9.170.4.155 SystemName()

```
const char* FCFSupport::System::SystemName ( ) const [inline]
```

Return the system name.

This is read from the system file.

References systemName.

#### 9.170.4.156 TheCar()

```
FCFSupport::Car* FCFSupport::System::TheCar (
    int i ) const [inline]
```

Get a car by index.

##### Parameters

<i>i</i>	The car's index.
----------	------------------

References cars, and i.

#### 9.170.4.157 TheCarGroup()

```
CarGroup* FCFSupport::System::TheCarGroup (
    int i ) const [inline]
```

Get a car class instance pointer given a car group index.

##### Parameters

<i>i</i>	The car group index.
----------	----------------------

References carGroups, i, and FCFSupport::CarGroup::MaxCarGroup.

#### 9.170.4.158 TheCarType()

```
CarType* FCFSupport::System::TheCarType (
    char c ) [inline]
```

Get a car type class instance pointer given a car type.

##### Parameters

<i>c</i>	The car type to lookup.
----------	-------------------------

References carTypes.

#### 9.170.4.159 TheDivision()

```
FCFSupport::Division* FCFSupport::System::TheDivision (
    int i ) [inline]
```

[Division](#) indexing function.

Warning: if the division at the specified index does not already exist, a new element is allocated with a NULL pointer.

##### Parameters

<i>i</i>	The division index to access.
----------	-------------------------------

References divisions, and i.

#### 9.170.4.160 TheIndustry()

```
FCFSupport::Industry* FCFSupport::System::TheIndustry (
    int i ) [inline]
```

[Industry](#) indexing function.

Warning: if the industry at the specified index does not already exist, a new element is allocated with a NULL pointer.

##### Parameters

<i>i</i>	The industry index to access.
----------	-------------------------------

References `i`, and industries.

#### 9.170.4.161 TheOwner()

```
Owner* FCFSupport::System::TheOwner (
    const char * initials ) [inline]
```

Get a car owner class instance pointer given a car owner's initials.

##### Parameters

<i>initials</i>	The car owner's initials.
-----------------	---------------------------

References owners.

#### 9.170.4.162 TheStation()

```
FCFSupport::Station* FCFSupport::System::TheStation (
    int i ) [inline]
```

`Station` indexing function.

Warning: if the station at the specified index does not already exist, a new element is allocated with a NULL pointer.

##### Parameters

<i>i</i>	The station index to access.
----------	------------------------------

References `FindStationByName()`, `i`, and stations.

#### 9.170.4.163 Today()

```
const string FCFSupport::System::Today ( ) const [private]
```

Return today's date.

**9.170.4.164 TotalCars()**

```
int FCFSupport::System::TotalCars ( ) const [inline]
```

Return the total number of cars.

References cars.

**9.170.4.165 TotalShifts()**

```
int FCFSupport::System::TotalShifts ( ) const [inline]
```

Return the total number of shifts.

References totalShifts.

**9.170.4.166 TrainByIndex()**

```
Train* FCFSupport::System::TrainByIndex (
    int i ) [inline]
```

[Train](#) indexing function.

Warning: if the train at the specified index does not already exist, a new element is allocated with a NULL pointer.

**Parameters**

<i>i</i>	The train index to access.
----------	----------------------------

References *i*, and trains.

**9.170.4.167 TrainByName()**

```
Train* FCFSupport::System::TrainByName (
    const char * name ) [inline]
```

[Train](#) indexing (by name) function.

Warning: if the train at the specified index (name) does not already exist, a new element is allocated with a NULL pointer.

## Parameters

<i>name</i>	<a href="#">Train</a> name to access.
-------------	---------------------------------------

References [trainIndex](#).

## 9.170.4.168 TrainCarPickupCheck()

```
bool FCFSupport::System::TrainCarPickupCheck (
    Car * car,
    Train * train,
    bool boxMove,
    CarVector & consist,
    bool & didAction,
    int Px,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer ) [private]
```

Check to see if we can really pick up this car.

## Parameters

<i>car</i>	The car to check.
<i>train</i>	The train to pick up the car for.
<i>boxMove</i>	Is this a box move?
<i>consist</i>	The train's consist.
<i>didAction</i>	Flag to set if something was done.
<i>Px</i>	The stop number that train is at.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.

## 9.170.4.169 TrainDropAllCars()

```
void FCFSupport::System::TrainDropAllCars (
    Train * train,
    int Px,
    CarVector & consist,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer ) [private]
```

Drop all cars from a train at the current stop (usually the last stop).

## Parameters

<i>train</i>	The train to drop cars from.
<i>Px</i>	The stop number that train is at.
<i>consist</i>	The train's consist.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.

## 9.170.4.170 TrainDropOneCar()

```
void FCFSupport::System::TrainDropOneCar (
    Car * car,
    Train * train,
    CarVector::iterator Lx,
    CarVector & consist,
    bool & didAction,
    int Px,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer ) [private]
```

Drop a single car.

## Parameters

<i>car</i>	The car to drop.
<i>train</i>	The train to drop the car from.
<i>Lx</i>	The index of the car to drop.
<i>consist</i>	The train's consist.
<i>didAction</i>	Flag to set if something was done.
<i>Px</i>	The stop number that train is at.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.

## 9.170.4.171 TrainIndex()

```
int FCFSupport::System::TrainIndex (
    const FCFSupport::Train * train ) const
```

Return a train's index.

## Parameters

<i>train</i>	The train to lookup.
--------------	----------------------

Referenced by StatsPeriod().

## 9.170.4.172 TrainLocalDrops()

```
void FCFSupport::System::TrainLocalDrops (
    Train * train,
    int Px,
    CarVector & consist,
    bool & didAction,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer ) [private]
```

Drop cars from a local (box move or way freight).

## Parameters

<i>train</i>	The train to drop cars from.
<i>Px</i>	The stop number that train is at.
<i>consist</i>	The train's consist.
<i>didAction</i>	Flag to set if something was done.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.

## 9.170.4.173 TrainLocalOriginate()

```
void FCFSupport::System::TrainLocalOriginate (
    Train * train,
    bool boxMove,
    int Px,
    CarVector & consist,
    bool & didAction,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer ) [private]
```

Make up a local train.

## Parameters

<i>train</i>	The train to make up.
--------------	-----------------------

## Parameters

<i>boxMove</i>	Is this a box move?
<i>Px</i>	The stop number that train is at.
<i>consist</i>	The train's consist.
<i>didAction</i>	Flag to set if something was done.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.

## 9.170.4.174 TrainLocalPickups()

```
void FCFSupport::System::TrainLocalPickups (
    Train * train,
    bool boxMove,
    int Px,
    CarVector & consist,
    bool & didAction,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer ) [private]
```

Pick up cars for a local train (box move or way freight).

## Parameters

<i>train</i>	The train to pick up cars for.
<i>boxMove</i>	Is this a box move?
<i>Px</i>	The stop number that train is at.
<i>consist</i>	The train's consist.
<i>didAction</i>	Flag to set if something was done.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.

## 9.170.4.175 TrainManifestDrops()

```
void FCFSupport::System::TrainManifestDrops (
    Train * train,
    int Px,
    CarVector & consist,
    bool & didAction,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer ) [private]
```

Drop cars from a manifest freight.



## Parameters

<i>train</i>	The train to drop cars from.
<i>Px</i>	The stop number that train is at.
<i>consist</i>	The train's consist.
<i>didAction</i>	Flag to set if something was done.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.

## 9.170.4.176 TrainManifestPickups()

```
void FCFSupport::System::TrainManifestPickups (
    Train * train,
    bool boxMove,
    int Px,
    CarVector & consist,
    bool & didAction,
    const FCFSupport::LogMessageCallback * Log,
    FCFSupport::PrinterDevice * printer ) [private]
```

Pick up cars for a manifest freight train.

## Parameters

<i>train</i>	The train to pick up cars for.
<i>boxMove</i>	Is this a box move?
<i>Px</i>	The stop number that train is at.
<i>consist</i>	The train's consist.
<i>didAction</i>	Flag to set if something was done.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.

## 9.170.4.177 TrainPickupOneCar()

```
void FCFSupport::System::TrainPickupOneCar (
    Car * car,
    Train * train,
    bool boxMove,
    CarVector & consist,
    bool & didAction,
    int Px,
    CarVector::iterator Lx,
```

```
const FCFSupport::LogMessageCallback * Log,  
FCFSupport::PrinterDevice * printer ) [private]
```

Pick up one car.

#### Parameters

<i>car</i>	The car to possibly pick up.
<i>train</i>	The train to pick up the car for.
<i>boxMove</i>	Is this a box move?
<i>consist</i>	The train's consist.
<i>didAction</i>	Flag to set if something was done.
<i>Px</i>	The stop number that train is at.
<i>Lx</i>	Place in the train to put the car if it is picked up.
<i>Log</i>	Log message callback.
<i>printer</i>	Printer device.

#### 9.170.4.178 TrainPrintConsistSummary()

```
void FCFSupport::System::TrainPrintConsistSummary (  
    Train * train,  
    CarVector & consist,  
    FCFSupport::PrinterDevice * printer ) [private]
```

Print a train's consist summary.

#### Parameters

<i>train</i>	The train to print a summary for.
<i>consist</i>	The train's consist.
<i>printer</i>	Printer device.

#### 9.170.4.179 TrainPrintFinalSummary()

```
void FCFSupport::System::TrainPrintFinalSummary (  
    Train * train,  
    FCFSupport::PrinterDevice * printer ) [private]
```

Print a train's final summary.

## Parameters

<i>train</i>	The train to print the final summary for.
<i>printer</i>	Printer device.

## 9.170.4.180 TrainPrintTown()

```
void FCFSupport::System::TrainPrintTown (
    const FCFSupport::Train * train,
    const FCFSupport::Station * curStation,
    FCFSupport::PrinterDevice * printer ) [private]
```

Print the town a train is in.

## Parameters

<i>train</i>	The train to print the town for.
<i>curStation</i>	The current station.
<i>printer</i>	Printer device.

## 9.170.4.181 TrainsFile()

```
const char* FCFSupport::System::TrainsFile ( ) const [inline]
```

Return the trains file's full path name.

References FCFSupport::PathName::FullPath(), and trainsFile.

## 9.170.4.182 trim()

```
string FCFSupport::System::trim (
    string line ) const [private]
```

Helper utility function to trim white space off the ends of a string.

## Parameters

<i>line</i>	The string to trim.
-------------	---------------------

#### 9.170.4.183 UpperCase()

```
const string FCFSupport::System::UpperCase (
    const string str ) const [private]
```

Convert a string to all uppercase letters.

##### Parameters

<i>str</i>	The string to convert.
------------	------------------------

#### 9.170.4.184 WriteOneCarToDisk()

```
bool FCFSupport::System::WriteOneCarToDisk (
    Car * car,
    ostream & stream ) [private]
```

Function to write one car to disk.

##### Parameters

<i>car</i>	The car to write.
<i>stream</i>	The output stream to write to.

### 9.170.5 Member Data Documentation

#### 9.170.5.1 carDest

```
Industry* FCFSupport::System::carDest [private]
```

A temporary for a car's location.

#### 9.170.5.2 carGroups

```
CarGroup* FCFSupport::System::carGroups[CarGroup::MaxCarGroup] [private]
```

Car group vector.

Referenced by TheCarGroup().

#### 9.170.5.3 carMovements

```
int FCFSupport::System::carMovements [private]
```

The number of cars movements.

Referenced by CarMovements().

#### 9.170.5.4 cars

```
CarVector FCFSupport::System::cars [private]
```

Car vector.

Referenced by AddCar(), NumberOfCars(), TheCar(), and TotalCars().

#### 9.170.5.5 carsAtDest

```
int FCFSupport::System::carsAtDest [private]
```

The number of cars at their destinations.

Referenced by CarsAtDest().

#### 9.170.5.6 carsAtDest\_carsInTransit

```
int FCFSupport::System::carsAtDest_carsInTransit [private]
```

The number of cars at their destinations and still in transit.

Referenced by CarsAtDest\_CarsInTransit().

**9.170.5.7 carsAtWorkBench**

```
int FCFSupport::System::carsAtWorkBench [private]
```

The number of cars at the workbench.

Referenced by CarsAtWorkBench().

**9.170.5.8 carsFile**

```
PathName FCFSupport::System::carsFile [private]
```

Full pathname of the cars file.

Referenced by CarsFile().

**9.170.5.9 carsInTransit**

```
int FCFSupport::System::carsInTransit [private]
```

The number of cars in transit.

Referenced by CarsInTransit().

**9.170.5.10 carsMoved**

```
int FCFSupport::System::carsMoved [private]
```

The number of cars moved.

Referenced by CarsMoved().

**9.170.5.11 carsMovedMore**

```
int FCFSupport::System::carsMovedMore [private]
```

The number of cars moved more then three times.

Referenced by CarsMovedMore().

#### 9.170.5.12 carsMovedOnce

```
int FCFSupport::System::carsMovedOnce [private]
```

The number of cars moved one time.

Referenced by CarsMovedOnce().

#### 9.170.5.13 carsMovedThree

```
int FCFSupport::System::carsMovedThree [private]
```

The number of cars moved three times.

Referenced by CarsMovedThree().

#### 9.170.5.14 carsMovedTwice

```
int FCFSupport::System::carsMovedTwice [private]
```

The number of cars moved two times.

Referenced by CarsMovedTwice().

#### 9.170.5.15 carsNotMoved

```
int FCFSupport::System::carsNotMoved [private]
```

The number of cars not moved.

Referenced by CarsNotMoved().

#### 9.170.5.16 carTypes

```
CarTypeMap FCFSupport::System::carTypes [private]
```

Car type map.

Referenced by FirstCarType(), LastCarType(), and TheCarType().

#### 9.170.5.17 carTypesFile

`PathName FCFSupport::System::carTypesFile [private]`

Full pathname of the car types file.

Referenced by `CarTypesFile()`.

#### 9.170.5.18 carTypesOrder

`char FCFSupport::System::carTypesOrder[CarType::MaxCarTypes] [private]`

`Car` type order vector.

Referenced by `CarTypesOrder()`.

#### 9.170.5.19 curDiv

`Division* FCFSupport::System::curDiv [private]`

Current division.

#### 9.170.5.20 deliver

`bool FCFSupport::System::deliver [private]`

Deliver flag.

#### 9.170.5.21 divisions

`DivisionMap FCFSupport::System::divisions [private]`

`Division` map.

Referenced by `FindDivisionByIndex()`, `FirstDivision()`, `LastDivision()`, `NumberOfDivisions()`, and `TheDivision()`.



#### 9.170.5.22 indScrapYard

```
const FCFSupport::Industry FCFSupport::System::indScrapYard [private]
```

The pointer to the scrapyard.

Referenced by IndScrapYard().

#### 9.170.5.23 industries

```
IndustryMap FCFSupport::System::industries [private]
```

Industries map.

Referenced by FindIndustryByIndex(), FirstIndustry(), IndRipTrack(), IndRipTrackConst(), LastIndustry(), NumberOfIndustries(), and TheIndustry().

#### 9.170.5.24 industriesFile

```
PathName FCFSupport::System::industriesFile [private]
```

Full pathname of the industries file.

Referenced by IndustriesFile().

#### 9.170.5.25 messageBuffer

```
char FCFSupport::System::messageBuffer[2048] [private]
```

Message buffer, used for error messages mostly.

#### 9.170.5.26 numberCars

```
int FCFSupport::System::numberCars [private]
```

The number of cars on a train.

**9.170.5.27 ordersFile**

`PathName` FCFSupport::System::ordersFile [private]

Full pathname of the train orders file.

Referenced by OrdersFile().

**9.170.5.28 originYard**

`Industry*` FCFSupport::System::originYard [private]

Origin Yard.

**9.170.5.29 owners**

`OwnerMap` FCFSupport::System::owners [private]

`Car` owner map.

Referenced by AddOwner(), FirstOwner(), LastOwner(), and TheOwner().

**9.170.5.30 ownersFile**

`PathName` FCFSupport::System::ownersFile [private]

Full pathname of the car owners file.

Referenced by OwnersFile().

**9.170.5.31 printAlpha**

`bool` FCFSupport::System::printAlpha [private]

Flag for printing alphabetical lists.

Referenced by PrintAlpha(), and SetPrintAlpha().

**9.170.5.32 printAtwice**

```
bool FCFSupport::System::printAtwice [private]
```

Flag for printing a second copy of alphabetical lists.

Referenced by PrintAtwice(), and SetPrintAtwice().

**9.170.5.33 printDispatch**

```
bool FCFSupport::System::printDispatch [private]
```

Flag for printing a dispatcher's report.

Referenced by PrintDispatch(), and SetPrintDispatch().

**9.170.5.34 printem**

```
bool FCFSupport::System::printem [private]
```

Flag for printing train movements.

Referenced by Printem(), and SetPrintem().

**9.170.5.35 printList**

```
bool FCFSupport::System::printList [private]
```

Flag for printing train switch lists.

Referenced by PrintList(), and SetPrintList().

**9.170.5.36 printLtwice**

```
bool FCFSupport::System::printLtwice [private]
```

Flag for printing a second copy of train switch lists.

Referenced by PrintLtwice(), and SetPrintLtwice().

**9.170.5.37 printYards**

```
bool FCFSupport::System::printYards [private]
```

Flag for printing yard switch lists.

Referenced by PrintYards(), and SetPrintYards().

**9.170.5.38 ranAllTrains**

```
int FCFSupport::System::ranAllTrains [private]
```

The ran all trains flag.

Referenced by RanAllTrains().

**9.170.5.39 sessionNumber**

```
int FCFSupport::System::sessionNumber [private]
```

Current session number.

Referenced by NextShift(), and SessionNumber().

**9.170.5.40 shiftNumber**

```
int FCFSupport::System::shiftNumber [private]
```

Current shift number.

Referenced by NextShift(), and ShiftNumber().

**9.170.5.41 stations**

```
StationMap FCFSupport::System::stations [private]
```

Station map.

Referenced by FirstStation(), LastStation(), NumberOfStations(), and TheStation().

#### 9.170.5.42 statsFile

`PathName FCFSupport::System::statsFile [private]`

Full pathname of the stats file.

Referenced by StatsFile().

#### 9.170.5.43 statsPeriod

`int FCFSupport::System::statsPeriod [private]`

The current stats period.

Referenced by StatsPeriod().

#### 9.170.5.44 switchList

`SwitchList FCFSupport::System::switchList [private]`

Switch lists.

#### 9.170.5.45 systemFile

`PathName FCFSupport::System::systemFile [private]`

Full pathname of the system file.

Referenced by SystemFile().

#### 9.170.5.46 systemName

`string FCFSupport::System::systemName [private]`

The system name.

Referenced by SystemName().

**9.170.5.47 totalLoads**

```
int FCFSupport::System::totalLoads [private]
```

The total number of loads.

**9.170.5.48 totalPickups**

```
int FCFSupport::System::totalPickups [private]
```

The total number of pickups.

**9.170.5.49 totalRevenueTons**

```
int FCFSupport::System::totalRevenueTons [private]
```

The total number of revenue tons.

**9.170.5.50 totalShifts**

```
int FCFSupport::System::totalShifts [private]
```

The total number of shifts.

Referenced by NextShift(), and TotalShifts().

**9.170.5.51 totalTons**

```
int FCFSupport::System::totalTons [private]
```

The total number of tons.

#### 9.170.5.52 trainEmpties

```
int FCFSupport::System::trainEmpties [private]
```

The number of empties on a train.

#### 9.170.5.53 trainIndex

```
TrainNameMap FCFSupport::System::trainIndex [private]
```

[Train](#) name map.

Referenced by [FindTrainByName\(\)](#), and [TrainByName\(\)](#).

#### 9.170.5.54 trainLastLocation

```
Industry* FCFSupport::System::trainLastLocation [private]
```

A trains last location.

#### 9.170.5.55 trainLength

```
int FCFSupport::System::trainLength [private]
```

[Train](#) length.

#### 9.170.5.56 trainLoads

```
int FCFSupport::System::trainLoads [private]
```

The number of loads on a train.

**9.170.5.57 trainLongest**

```
int FCFSupport::System::trainLongest [private]
```

The longest a train has been.

**9.170.5.58 trainPrintOK**

```
bool FCFSupport::System::trainPrintOK [private]
```

[Train](#) print flag.

**9.170.5.59 trains**

```
TrainMap FCFSupport::System::trains [private]
```

[Train](#) map.

Referenced by FindTrainByIndex(), FirstTrain(), LastTrain(), NumberOfTrains(), and TrainByIndex().

**9.170.5.60 trainsFile**

```
PathName FCFSupport::System::trainsFile [private]
```

Full pathname of the trains file.

Referenced by TrainsFile().

**9.170.5.61 trainTons**

```
int FCFSupport::System::trainTons [private]
```

The number of tons on a train.



## 9.170.5.62 wayFreight

```
bool FCFSupport::System::wayFreight [private]
```

Way freight flag.

## 9.170.5.63 whitespace

```
const string FCFSupport::System::whitespace [static], [private]
```

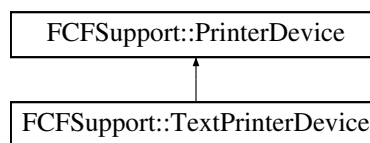
String of white space characters.

## 9.171 FCFSupport::TextPrinterDevice Class Reference

Derived class for printing on generic plain text printers.

```
#include <TextPrinter.h>
```

Inheritance diagram for FCFSupport::TextPrinterDevice:



## Public Member Functions

- [TextPrinterDevice](#) (const string filename="", const string title="", [PageSize pageSize=Letter](#), char \*\*outmessage=NULL)  
*Constructor.*
- virtual bool [OpenPrinter](#) (const string filename, [PageSize pageSize=Letter](#), char \*\*outmessage=NULL)  
*Member function to open the printer.*
- virtual bool [ClosePrinter](#) (char \*\*outmessage)  
*Close the printer.*
- virtual bool [NewPage](#) (const string heading="")  
*Perform a page feed and print a heading.*
- virtual bool [PutLine](#) (const string line)  
*Print out a string and follow it with a new line sequence.*
- virtual bool [Put](#) (const string text)  
*Print a string of text.*
- virtual bool [Tab](#) (int column)  
*Tab over to the specified column.*
- virtual [~TextPrinterDevice](#) ()  
*Destructor.*

## Private Attributes

- ofstream [printerStream](#)  
*Output stream.*
- int [currentColumn](#)  
*Current column.*

## Additional Inherited Members

### 9.171.1 Detailed Description

Derived class for printing on generic plain text printers.

Only the tab columns and pagination (every 60 lines or no) is implemented, that is, a very basic plain text output device.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.171.2 Constructor & Destructor Documentation

#### 9.171.2.1 TextPrinterDevice()

```
FCFSupport::TextPrinterDevice::TextPrinterDevice (
    const string filename = "",
    const string title = "",
    PageSize pageSize = Letter,
    char ** outmessage = NULL )
```

Constructor.

Create a new printer device instance from a set of parameters, all of which have default values, so this also doubles as the default base constructor.

#### Parameters

<i>filename</i>	Output filename.
<i>title</i>	An internal document title string.
<i>pageSize</i>	The page size to use.
<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur. This parameter is hidden from the Tcl interface.

### 9.171.2.2 ~TextPrinterDevice()

```
virtual FCFSupport::TextPrinterDevice::~~TextPrinterDevice ( ) [virtual]
```

Destructor.

Close the printer.

## 9.171.3 Member Function Documentation

### 9.171.3.1 ClosePrinter()

```
virtual bool FCFSupport::TextPrinterDevice::ClosePrinter (
    char ** outmessage ) [virtual]
```

Close the printer.

#### Parameters

<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur. This parameter is hidden from the Tcl interface.
-------------------	--

Reimplemented from [FCFSupport::PrinterDevice](#).

### 9.171.3.2 NewPage()

```
virtual bool FCFSupport::TextPrinterDevice::NewPage (
    const string heading = "" ) [virtual]
```

Perform a page feed and print a heading.

#### Parameters

<i>heading</i>	The heading string.
----------------	---------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

### 9.171.3.3 OpenPrinter()

```
virtual bool FCFSupport::TextPrinterDevice::OpenPrinter (
    const string filename,
    PageSize pageSize = Letter,
    char ** outmessage = NULL ) [virtual]
```

Member function to open the printer.

#### Parameters

<i>filename</i>	Output filename.
<i>pageSize</i>	The page size to use.
<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur. This parameter is hidden from the Tcl interface.

Reimplemented from [FCFSupport::PrinterDevice](#).

### 9.171.3.4 Put()

```
virtual bool FCFSupport::TextPrinterDevice::Put (
    const string text ) [virtual]
```

Print a string of text.

Don't include a newline.

#### Parameters

<i>text</i>	The string to print.
-------------	----------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

### 9.171.3.5 PutLine()

```
virtual bool FCFSupport::TextPrinterDevice::PutLine (
    const string line ) [virtual]
```

Print out a string and follow it with a new line sequence.

#### Parameters

<i>line</i>	The line to print.
-------------	--------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

#### 9.171.3.6 Tab()

```
virtual bool FCFSupport::TextPrinterDevice::Tab (  
    int column ) [virtual]
```

Tab over to the specified column.

##### Parameters

<i>column</i>	The desired tab column.
---------------	-------------------------

Reimplemented from [FCFSupport::PrinterDevice](#).

### 9.171.4 Member Data Documentation

#### 9.171.4.1 currentColumn

```
int FCFSupport::TextPrinterDevice::currentColumn [private]
```

Current column.

#### 9.171.4.2 printerStream

```
ofstream FCFSupport::TextPrinterDevice::printerStream [private]
```

Output stream.

## 9.172 CTCPanel::ThreeWaySW Class Reference

Three Way Switch (turnout) object type.

## Public Member Functions

- [ThreeWaySW](#) (name, \_ctcpanel, \_canvas,...)  
*Construct a [ThreeWaySW](#) object.*
- [~ThreeWaySW](#) ()  
*Clean up all data objects and free up all resources.*
- [getv](#) ()  
*Method to get our value (state).*
- [setv](#) (value)  
*Method to set out value (state).*
- [geti](#) (ind)  
*Method to get the state of one of our indicators (none).*
- [seti](#) (ind, value)  
*Method to set an indicator's state (none).*
- [invoke](#) ()  
*Method to invoke the switch.*

## Private Member Functions

- [\\_configureLabel](#) (option, value)  
*Method to update the label option.*

## Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*
- [state](#)  
*The state of the points.*

### 9.172.1 Detailed Description

Three Way Switch (turnout) object type.

These are on the schematic and represent a switch on the Schematic.

#### Parameters

<a href="#">_ctcpanel</a>	The <a href="#">CTCPanel</a> megawidget.
<a href="#">_canvas</a>	The schematic canvas to draw the switch on.

## Parameters

...	<p>Options:</p> <ul style="list-style-type: none"> <li>• -x The x coordinate of the object (readonly, default 0).</li> <li>• -y The y coordinate of the object (readonly, default 0).</li> <li>• -controlpoint The name of the control point this label is part of (readonly, default CP1).</li> <li>• -label The label of the switch (default "1").</li> <li>• -orientation The orientation (8-way) of the switch (readonly, default 0).</li> <li>• -flipped Whether or not the switch is flipped (readonly, default no).</li> <li>• -statecommand A command to run to get the switch's state (default {}).</li> <li>• -occupiedcommand A command to run to find out if the switch is occupied (default {}).</li> </ul>
-----	--

Defined coords terminals:

- Common Points.
- Main Mainline.
- LDivergence Left branch.
- RDivergence Right branch.

Defined values (states):

- Normal Points are aligned for the mainline.
- Right Points are aligned for the Right branch.
- Left Points are aligned for the Left branch.
- Unknown Point are not aligned for any route (eg the points are in motion).

Defined indicators: none.

## Author

Robert Heller <heller@deepsoft.com>

## 9.172.2 Constructor &amp; Destructor Documentation

## 9.172.2.1 ThreeWaySW()

```
CTCPanel::ThreeWaySW::ThreeWaySW (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a [ThreeWaySW](#) object.

## Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the <a href="#">ThreeWaySW</a> on.
<code>...</code>	Option list.

**9.172.2.2** `~ThreeWaySW()`

```
CTCPanel::ThreeWaySW::~~ThreeWaySW ( )
```

Clean up all data objects and free up all resources.

**9.172.3** Member Function Documentation**9.172.3.1** `_configureLabel()`

```
CTCPanel::ThreeWaySW::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

**9.172.3.2** `geti()`

```
CTCPanel::ThreeWaySW::geti (
    ind )
```

Method to get the state of one of our indicators (none).

**9.172.3.3** `getv()`

```
CTCPanel::ThreeWaySW::getv ( )
```

Method to get our value (state).



#### 9.172.3.4 invoke()

```
CTCPanel::ThreeWaySW::invoke ( )
```

Method to invoke the switch.

#### 9.172.3.5 seti()

```
CTCPanel::ThreeWaySW::seti (
    ind ,
    value )
```

Method to set an indicator's state (none).

#### 9.172.3.6 setv()

```
CTCPanel::ThreeWaySW::setv (
    value )
```

Method to set out value (state).

##### Parameters

<i>value</i>	The new state to set.
--------------	-----------------------

### 9.172.4 Member Data Documentation

#### 9.172.4.1 canvas

```
CTCPanel::ThreeWaySW::canvas [private]
```

The canvas component (parent widget component).

#### 9.172.4.2 ctcpanel

CTCPanel::ThreeWaySW::ctcpanel [private]

The CTC Panel component (parent widget).

#### 9.172.4.3 state

CTCPanel::ThreeWaySW::state [private]

The state of the points.

### 9.173 CTCPanel::ThroughYard Class Reference

Through Yard object type.

#### Public Member Functions

- [ThroughYard](#) (name, \_ctcpanel, \_canvas,...)  
*Construct a [ThroughYard](#) object.*
- [~ThroughYard](#) ()  
*Clean up all data objects and free up all resources.*
- [setv](#) (value)  
*Method to set out value (state).*
- [geti](#) (ind)  
*Method to get the state of one of our indicators (none).*
- [seti](#) (ind, value)  
*Method to set an indicator's state (none).*
- [invoke](#) ()  
*Method to invoke the [ThroughYard](#).*

#### Private Member Functions

- [\\_configureLabel](#) (option, value)  
*Method to update the label option.*

#### Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*

## Static Private Attributes

- static [\\_ThroughYard\\_Poly](#)  
*Polygon coordinates for a through yard.*

### 9.173.1 Detailed Description

Through Yard object type.

These are on the schematic and represent a piece of track on the Schematic.

#### Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the <a href="#">ThroughYard</a> on.
<code>...</code>	Options: <ul style="list-style-type: none"> <li>• <code>-x</code> The x coordinate of the object (readonly, default 0).</li> <li>• <code>-y</code> The y coordinate of the object (readonly, default 0).</li> <li>• <code>-controlpoint</code> The name of the control point this label is part of (readonly, default Yard).</li> <li>• <code>-label</code> The label of the <a href="#">ThroughYard</a> (default "1").</li> <li>• <code>-position</code> The position of the label (readonly, default below).</li> <li>• <code>-orientation</code> The orientation (8-way) (readonly, default 0).</li> <li>• <code>-flipped</code> Whether the yard is flipped (readonly, default no).</li> <li>• <code>-occupiedcommand</code> A command to run to find out if the <a href="#">ThroughYard</a> is occupied (default {}).</li> </ul>

Defined coords terminals:

- EntryL Left yard throat.
- EntryR Right yard throat.

Defined values (states): none. Defined indicators: none.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.173.2 Constructor & Destructor Documentation

### 9.173.2.1 ThroughYard()

```
CTCPanel::ThroughYard::ThroughYard (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a [ThroughYard](#) object.

#### Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The schematic canvas to draw the <a href="#">ThroughYard</a> on.
<code>...</code>	Option list.

### 9.173.2.2 ~ThroughYard()

```
CTCPanel::ThroughYard::~~ThroughYard ( )
```

Clean up all data objects and free up all resources.

## 9.173.3 Member Function Documentation

### 9.173.3.1 \_configureLabel()

```
CTCPanel::ThroughYard::_configureLabel (
    option ,
    value ) [private]
```

Method to update the label option.

### 9.173.3.2 geti()

```
CTCPanel::ThroughYard::geti (
    ind )
```

Method to get the state of one of our indicators (none).

#### 9.173.3.3 invoke()

```
CTCPanel::ThroughYard::invoke ( )
```

Method to invoke the [ThroughYard](#).

#### 9.173.3.4 seti()

```
CTCPanel::ThroughYard::seti (
    ind ,
    value )
```

Method to set an indicator's state (none).

#### 9.173.3.5 setv()

```
CTCPanel::ThroughYard::setv (
    value )
```

Method to set out value (state).

##### Parameters

<i>value</i>	The new state to set.
--------------	-----------------------

### 9.173.4 Member Data Documentation

#### 9.173.4.1 \_ThroughYard\_Poly

```
CTCPanel::ThroughYard::_ThroughYard_Poly [static], [private]
```

Polygon coordinates for a through yard.

#### 9.173.4.2 canvas

CTCPanel::ThroughYard::canvas [private]

The canvas component (parent widget component).

#### 9.173.4.3 ctcpnl

CTCPanel::ThroughYard::ctcpnl [private]

The CTC Panel component (parent widget).

## 9.174 TTSupport::TimeRange Class Reference

The [TimeRange](#) class implements a range of times.

```
#include <Station.h>
```

### Public Member Functions

- [TimeRange](#) (double from\_<sub>=</sub>0.0, double to\_<sub>=</sub>0.0)  
*Construct a time range, from a start and end time.*
- double [From](#) () const  
*Return the low end of the range.*
- double [To](#) () const  
*Return the high end of the range.*
- bool [ContainsTime](#) (double time) const  
*Does this interval contain the specified time?*
- bool [operator<](#) (const [TimeRange](#) &other) const  
*Less than operator.*
- bool [operator>](#) (const [TimeRange](#) &other) const  
*Greater than operator.*
- bool [operator==](#) (const [TimeRange](#) &other) const  
*Equality to operator.*
- bool [operator<=](#) (const [TimeRange](#) &other) const  
*Less than or equal operator.*
- bool [operator>=](#) (const [TimeRange](#) &other) const  
*Greater than or equal operator.*
- [TimeRange](#) (const [TimeRange](#) &other)  
*Copy constructor: create a clone of a [TimeRange](#).*
- [TimeRange](#) & [operator=](#) (const [TimeRange](#) &other)  
*Assign a [TimeRange](#) to another [TimeRange](#).*
- ostream & [Write](#) (ostream &stream) const  
*Write ourselves to an output stream.*
- istream & [Read](#) (istream &stream)  
*Read ourselves from an input stream.*

## Private Attributes

- double [from](#)  
*Start time.*
- double [to](#)  
*End time.*

### 9.174.1 Detailed Description

The [TimeRange](#) class implements a range of times.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.174.2 Constructor & Destructor Documentation

#### 9.174.2.1 TimeRange() [1/2]

```
TTSupport::TimeRange::TimeRange (
    double from_ = 0.0,
    double to_ = 0.0 ) [inline]
```

Construct a time range, from a start and end time.

#### Parameters

<i>from_</i>	The start time.
<i>to_</i>	The end time.

References [TTSupport::Occupied::from](#).

#### 9.174.2.2 TimeRange() [2/2]

```
TTSupport::TimeRange::TimeRange (
    const TimeRange & other ) [inline]
```

Copy constructor: create a clone of a [TimeRange](#).

**Parameters**

<i>other</i>	The other <a href="#">TimeRange</a> object.
--------------	---

References [TTSupport::Occupied::from](#), [from](#), and [to](#).

### 9.174.3 Member Function Documentation

#### 9.174.3.1 ContainsTime()

```
bool TTSupport::TimeRange::ContainsTime (
    double time ) const [inline]
```

Does this interval contain the specified time?

**Parameters**

<i>time</i>	The time to check for.
-------------	------------------------

References [TTSupport::Occupied::from](#).

#### 9.174.3.2 From()

```
double TTSupport::TimeRange::From ( ) const [inline]
```

Return the low end of the range.

References [TTSupport::Occupied::from](#).

#### 9.174.3.3 operator<()

```
bool TTSupport::TimeRange::operator< (
    const TimeRange & other ) const [inline]
```

Less than operator.



## Parameters

<i>other</i>	The time range to compare to.
--------------	-------------------------------

References from.

## 9.174.3.4 operator&lt;=()

```
bool TTSupport::TimeRange::operator<= (
    const TimeRange & other ) const [inline]
```

Less than or equal operator.

## Parameters

<i>other</i>	The time range to compare to.
--------------	-------------------------------

## 9.174.3.5 operator=()

```
TimeRange& TTSupport::TimeRange::operator= (
    const TimeRange & other ) [inline]
```

Assign a [TimeRange](#) to another [TimeRange](#).

## Parameters

<i>other</i>	The other <a href="#">TimeRange</a> object.
--------------	---

References [TTSupport::Occupied::from](#), [from](#), [TTSupport::Occupied::Read\(\)](#), [Icc::stream](#), [to](#), and [TTSupport::Occupied::Write\(\)](#).

## 9.174.3.6 operator==()

```
bool TTSupport::TimeRange::operator== (
    const TimeRange & other ) const [inline]
```

Equality to operator.

**Parameters**

<i>other</i>	The time range to compare to.
--------------	-------------------------------

References TTSupport::Occupied::from, from, and to.

**9.174.3.7 operator>()**

```
bool TTSupport::TimeRange::operator> (
    const TimeRange & other ) const [inline]
```

Greater than operator.

**Parameters**

<i>other</i>	The time range to compare to.
--------------	-------------------------------

References TTSupport::Occupied::from, and to.

**9.174.3.8 operator>=()**

```
bool TTSupport::TimeRange::operator>= (
    const TimeRange & other ) const [inline]
```

Greater than or equal operator.

**Parameters**

<i>other</i>	The time range to compare to.
--------------	-------------------------------

**9.174.3.9 Read()**

```
istream& TTSupport::TimeRange::Read (
    istream & stream )
```

Read ourselves from an input stream.

## Parameters

<i>stream</i>	The stream to read from.
---------------	--------------------------

## 9.174.3.10 To()

```
double TTSupport::TimeRange::To ( ) const [inline]
```

Return the high end of the range.

## 9.174.3.11 Write()

```
ostream& TTSupport::TimeRange::Write (
    ostream & stream ) const
```

Write ourselves to an output stream.

## Parameters

<i>stream</i>	The stream to write to.
---------------	-------------------------

## 9.174.4 Member Data Documentation

## 9.174.4.1 from

```
double TTSupport::TimeRange::from [private]
```

Start time.

Referenced by operator<(), operator=(), operator==(), and TimeRange().

## 9.174.4.2 to

```
double TTSupport::TimeRange::to [private]
```

End time.

Referenced by operator=(), operator==(), operator>(), and TimeRange().

## 9.175 TTSupport::TimeTableSystem Class Reference

This is the main Time Table Class.

```
#include <TimeTableSystem.h>
```

### Public Member Functions

- [TimeTableSystem](#) (const string filename, char \*\*outmessage=NULL)  
*The constructor that creates a time table system from an existing file.*
- [TimeTableSystem](#) (const string name, int timescale, int timeinterval)  
*The constructor that creates a new, empty time table system from scratch, given a set of esential parameters.*
- [~TimeTableSystem](#) ()  
*Destructor.*
- int [AddStation](#) (string name, double smile)  
*Add a new station to the system.*
- int [FindStationByName](#) (string name)  
*Find a station by name.*
- int [NumberOfStations](#) () const  
*Number of stations.*
- [TTSupport::Station \\* lthStation](#) (int i)  
*Return lth station object.*
- const char \* [StationName](#) (int i) const  
*Return the lth station name.*
- double [SMile](#) (int i) const  
*Return the lth station's scale mile location.*
- double [TotalLength](#) () const  
*The total length of the route in scale miles.*
- int [DuplicateStationIndex](#) (int i) const  
*The duplicate station index for a given station.*
- void [SetDuplicateStationIndex](#) (int i, int dup)  
*Set the duplicate station index for a given station.*
- [StorageTrack \\* AddStorageTrack](#) (int i, string name)  
*Add a storage track to a station.*
- [StorageTrack \\* FindStorageTrack](#) (int i, string name)  
*Find a storage track at a station.*
- [Cab \\* AddCab](#) (string name, string color)  
*Add a new cab to the system.*
- int [NumberOfCabs](#) () const  
*The nymber of cabs.*
- [Train \\* AddTrain](#) (string name, string number, int speed, int classnumber, int departure, int start=0, int end=-1)  
*Add a train to the system, short version.*
- [Train \\* AddTrainLongVersion](#) (string name, string number, int speed, int classnumber, int departure, int start, int end, const [doubleVector](#) \*layoverVector, const [stringVector](#) \*cabnameVector, const [stringVector](#) \*storageTrack↔Vector, char \*\*outmessage=NULL)  
*Add a train to the system, long version (includes storage track checking).*

- bool [DeleteTrain](#) (string number, char \*\*outmessage=NULL)  
*Delete a train.*
- Cab \* [FindCab](#) (string name) const  
*Find a cab (by name).*
- Train \* [FindTrainByName](#) (string name) const  
*Find a train by name.*
- Train \* [FindTrainByNumber](#) (string number) const  
*Find a train by number (or symbol).*
- int [NumberOfTrains](#) () const  
*Return the number of trains.*
- int [NumberOfNotes](#) () const  
*Return the number of notes.*
- const char \* [Note](#) (int i)  
*Return the ith note (1-based!) as a string.*
- int [AddNote](#) (string newnote)  
*Add a note to the notes vector.*
- bool [SetNote](#) (int i, string note)  
*Set the ith note (1-based!).*
- const char \* [GetPrintOption](#) (const char \*key) const  
*Fetch a print option.*
- void [SetPrintOption](#) (const char \*key, string value)  
*Set a print option.*
- bool [WriteNewTimeTableFile](#) (string filename="TimeTableFile.tt", bool setfilename=false, char \*\*outmessage=NULL)  
*Write out a Time Table System to a new file.*
- bool [WriteOldTimeTableFile](#) (char \*\*outmessage=NULL)  
*Write an old time table file.*
- int [TimeScale](#) () const  
*Return time scale.*
- int [TimeInterval](#) () const  
*Return time interval.*
- const char \* [Name](#) () const  
*Return the name of the system.*
- const char \* [Filename](#) () const  
*Return file pathname.*
- bool [CreateLaTeXTimetable](#) (string filename, char \*\*outmessage=NULL)  
*Create a LaTeX file for generating a (hard copy) Employee Timetable.*
- CabNameMap::const\_iterator [FirstCab](#) () const  
*First cab.*
- CabNameMap::const\_iterator [LastCab](#) () const  
*Last cab.*
- TrainNumberMap::const\_iterator [FirstTrain](#) () const  
*First train.*
- TrainNumberMap::const\_iterator [LastTrain](#) () const  
*Last train.*
- OptionHashMap::const\_iterator [FirstPrintOption](#) () const  
*First Print option.*
- OptionHashMap::const\_iterator [LastPrintOption](#) () const  
*Last Print option.*

## Protected Member Functions

- [TimeTableSystem](#) ()  
*The default constructor.*

## Private Member Functions

- string [ReadNote](#) (istream &in) const  
*Read in a note.*
- ostream & [WriteNote](#) (ostream &out, string note) const  
*Write out a note.*
- bool [MakeTimeTableGroupByClass](#) (ostream &out, [TrainList](#) &allTrains, [TrainList](#) &forwardTrains, [TrainList](#) &backwardTrains, char \*\*outmessage=NULL)  
*Make a time table grouped by class.*
- bool [MakeTimeTableGroupManually](#) (ostream &out, int maxTrains, [TrainList](#) &allTrains, [TrainList](#) &forwardTrains, [TrainList](#) &backwardTrains, char \*\*outmessage=NULL)  
*Make a time table grouped manually.*
- bool [MakeTimeTableOneTable](#) (ostream &out, [TrainList](#) &allTrains, [TrainList](#) &forwardTrains, [TrainList](#) &backwardTrains, string header, string sectionTOP, char \*\*outmessage=NULL)  
*Make a time table as a single table.*
- bool [MakeTimeTableOneTableStationsLeft](#) (ostream &out, [TrainList](#) &trains, string header, string sectionTOP, char \*\*outmessage=NULL)  
*Make a time table as a single table, with the stations on the left (single direction trains).*
- bool [MakeTimeTableOneTableStationsCenter](#) (ostream &out, [TrainList](#) &forwardTrains, [TrainList](#) &backwardTrains, string header, string sectionTOP, char \*\*outmessage=NULL)  
*Make a time table as a single table, with the stations in the center (bi-directional trains).*
- void [ComputeTimes](#) ([TrainTimesAtStation](#) &timesAtStations, [TrainList](#) &trains)  
*Precompute station times, given a list of trains.*

## Private Attributes

- string [name](#)  
*The name of the time table system.*
- [PathName](#) [filepath](#)  
*The pathname of the file the system was loaded from.*
- int [timescale](#)  
*Time scale.*
- int [timeinterval](#)  
*Time interval.*
- [StationVector](#) [stations](#)  
*Station stop vector.*
- [CabNameMap](#) [cabs](#)  
*Cap name map.*
- [TrainNumberMap](#) [trains](#)  
*Train number/symbol map.*
- vector< string > [notes](#)

*Notes.*

- [OptionHashMap printOptions](#)

*Print option hash table.*

- bool [TOCP](#)

*Table Of Contents?*

- string [DirectionName](#)

*Direction Name.*

### 9.175.1 Detailed Description

This is the main Time Table Class.

It implements all of the basic data and algorithms used in the Time Table program.

This class includes code to load a set of stations and the trains that run between these stations, along with code to read and write a time table file and code to create a formatted time table, suitable for printing (by way of LaTeX).

#### Author

Robert Heller <heller@deepsoft.com>

### 9.175.2 Constructor & Destructor Documentation

#### 9.175.2.1 TimeTableSystem() [1/3]

```
TTSupport::TimeTableSystem::TimeTableSystem ( ) [inline], [protected]
```

The default constructor.

This is protected to prevent the creation of an uninitialized class instance, which would be an error. Making the default constructor a protected method will cause a compiler error when application code attempts to create a [TimeTableSystem](#) instance without using one of the parameterized constructor. There is no meaningful way to create a proper [TimeTableSystem](#) without supplying some parameters.

#### 9.175.2.2 TimeTableSystem() [2/3]

```
TTSupport::TimeTableSystem::TimeTableSystem (
    const string filename,
    char ** outmessage = NULL )
```

The constructor that creates a time table system from an existing file.

The file is read in and the class is properly initialized from the data in the file.

## Parameters

<i>filename</i>	The name of the file to load.
<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur.

## 9.175.2.3 TimeTableSystem() [3/3]

```
TTSupport::TimeTableSystem::TimeTableSystem (
    const string name,
    int timescale,
    int timeinterval )
```

The constructor that creates a new, empty time table system from scratch, given a set of esential parameters.

## Parameters

<i>name</i>	The name of the time table system.
<i>timescale</i>	Number of time units per 24 hours. There are 1440 minutes in 24 hours.
<i>timeinterval</i>	The tick frequency in time units.

## 9.175.2.4 ~TimeTableSystem()

```
TTSupport::TimeTableSystem::~~TimeTableSystem ( )
```

Destructor.

Properly clean up and free up all used space.

## 9.175.3 Member Function Documentation

## 9.175.3.1 AddCab()

```
Cab* TTSupport::TimeTableSystem::AddCab (
    string name,
    string color )
```

Add a new cab to the system.

With DC systems this would be an actual cab. With DCC systems, this can be used to define a logical operator for the train. The color is used for visual distintion. A pointer to the new cab object is returned.



## Parameters

<i>name</i>	The name of the cab.
<i>color</i>	The color of the cab.

## 9.175.3.2 AddNote()

```
int TTSupport::TimeTableSystem::AddNote (
    string newnote ) [inline]
```

Add a note to the notes vector.

## Parameters

<i>newnote</i>	The text of the new note.
----------------	---------------------------

## 9.175.3.3 AddStation()

```
int TTSupport::TimeTableSystem::AddStation (
    string name,
    double smile )
```

Add a new station to the system.

Creates a new [Station](#) class instance and adds it to the station vector. Stations must be added in order of their scale mile location. If the new station is out of order, -1 is returned and the station is not added!

## Parameters

<i>name</i>	The name of the station.
<i>smile</i>	The scale mile along the route where the station is located.

## 9.175.3.4 AddStorageTrack()

```
StorageTrack* TTSupport::TimeTableSystem::AddStorageTrack (
    int i,
    string name ) [inline]
```

Add a storage track to a station.

Sometimes stations, especially major terminals, have extra tracks for storing terminating and originating trains. Returns the NULL pointer if the index is out of range. Otherwise returns the pointer to the new [StorageTrack](#) object.

#### Parameters

<i>i</i>	The index of the station to be updated.
<i>name</i>	The name for the new storage track.

References [i](#).

#### 9.175.3.5 AddTrain()

```
Train* TTSupport::TimeTableSystem::AddTrain (
    string name,
    string number,
    int speed,
    int classnumber,
    int departure,
    int start = 0,
    int end = -1 )
```

Add a train to the system, short version.

Creates a new [Train](#) object and adds it to the train map. The short version assumes that the train does not layover at any of the stops. Layover times can be added later. Returns a pointer to the new [Train](#) object.

#### Parameters

<i>name</i>	The name of the train.
<i>number</i>	The number (or symbol) of the train.
<i>speed</i>	The trains maximum speed.
<i>classnumber</i>	The class (inverse priority) of the train.
<i>departure</i>	The train's departure time.
<i>start</i>	The train's origin station index. Defaults to the first station.
<i>end</i>	The train's destination station index. Defaults to the last station.

#### 9.175.3.6 AddTrainLongVersion()

```
Train* TTSupport::TimeTableSystem::AddTrainLongVersion (
    string name,
    string number,
    int speed,
```

```

    int classnumber,
    int departure,
    int start,
    int end,
    const doubleVector * layoverVector,
    const stringVector * cabnameVector,
    const stringVector * storageTrackVector,
    char ** outmessage = NULL )

```

Add a train to the system, long version (includes storage track checking).

This version includes layover times, cabnames, and storage track assignments. Returns a pointer to the new [Train](#) object or the NULL pointer if there was an error, in which case the error message will be stored in the pointer provided.

#### Parameters

<i>name</i>	The name of the train.
<i>number</i>	The number (or symbol) of the train.
<i>speed</i>	The trains maximum speed.
<i>classnumber</i>	The class (inverse priority) of the train.
<i>departure</i>	The train's departure time.
<i>start</i>	The train's origin station index.
<i>end</i>	The train's destination station index.
<i>layoverVector</i>	The train's layover vector.
<i>cabnameVector</i>	The train's departure cab name vector.
<i>storageTrackVector</i>	The train's storage track name vector.
<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur.

#### 9.175.3.7 ComputeTimes()

```

void TTSupport::TimeTableSystem::ComputeTimes (
    TrainTimesAtStation & timesAtStations,
    TrainList & trains ) [private]

```

Precompute station times, given a list of trains.

This helper function creates the table cell information for a time table.

#### Parameters

<i>timesAtStations</i>	The time table matrix to be filled in.
<i>trains</i>	A list of trains to process.

### 9.175.3.8 CreateLaTeXTimetable()

```
bool TTSupport::TimeTableSystem::CreateLaTeXTimetable (
    string filename,
    char ** outmessage = NULL )
```

Create a LaTeX file for generating a (hard copy) Employee Timetable.

This method create a LaTeX source file from the information in the time table structure. It access various print options to control how the LaTeX file is generated.

#### Parameters

<i>filename</i>	The name of the LaTeX file to create.
<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur.

### 9.175.3.9 DeleteTrain()

```
bool TTSupport::TimeTableSystem::DeleteTrain (
    string number,
    char ** outmessage = NULL )
```

Delete a train.

Returns true if the train was successfully deleted and false if not. If the train was not deleted, an error message will be provided in the pointer provided.

#### Parameters

<i>number</i>	The train number or symbol.
<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur.

### 9.175.3.10 DuplicateStationIndex()

```
int TTSupport::TimeTableSystem::DuplicateStationIndex (
    int i ) const [inline]
```

The duplicate station index for a given station.

Only meaningful for out and back type layouts or layouts that have shared trackage. This would be stations along shared trackage. Returns -1 if the index is out of range or if there is not a duplicate station for the ith station.

**Parameters**

<i>i</i>	The index of the station.
----------	---------------------------

References [i](#).

**9.175.3.11 Filename()**

```
const char* TTSupport::TimeTableSystem::Filename ( ) const [inline]
```

Return file pathname.

**9.175.3.12 FindCab()**

```
Cab* TTSupport::TimeTableSystem::FindCab (
    string name ) const
```

Find a cab (by name).

Returns the pointer to the named cab or NULL if the cab was not found.

**Parameters**

<i>name</i>	The cab name to look for.
-------------	---------------------------

**9.175.3.13 FindStationByName()**

```
int TTSupport::TimeTableSystem::FindStationByName (
    string name )
```

Find a station by name.

Returns the index of the station or -1 if the station cannot be found.

**Parameters**

<i>name</i>	The name of the station.
-------------	--------------------------

#### 9.175.3.14 FindStorageTrack()

```
StorageTrack* TTSupport::TimeTableSystem::FindStorageTrack (
    int i,
    string name ) [inline]
```

Find a storage track at a station.

Sometimes stations, especially major terminals, have extra tracks for storing terminating and originating trains. Returns the NULL pointer if the index is out of range or if there is no storage track with the specified name. Otherwise the [StorageTrack](#) object pointer is returned.

##### Parameters

<i>i</i>	The index of the station to be updated.
<i>name</i>	The name of the storage track.

References [i](#).

#### 9.175.3.15 FindTrainByName()

```
Train* TTSupport::TimeTableSystem::FindTrainByName (
    string name ) const
```

Find a train by name.

Returns the pointer to the named train or NULL if the train was not found.

##### Parameters

<i>name</i>	The train name to look for.
-------------	-----------------------------

#### 9.175.3.16 FindTrainByNumber()

```
Train* TTSupport::TimeTableSystem::FindTrainByNumber (
    string number ) const
```

Find a train by number (or symbol).

Returns the pointer to the train or NULL if the train was not found.

**Parameters**

<i>number</i>	The train number (or symbol) to look for.
---------------	---

**9.175.3.17 FirstCab()**

```
CabNameMap::const_iterator TTSupport::TimeTableSystem::FirstCab ( ) const [inline]
```

First cab.

Return a const iterator for the first cab.

**9.175.3.18 FirstPrintOption()**

```
OptionHashMap::const_iterator TTSupport::TimeTableSystem::FirstPrintOption ( ) const [inline]
```

First Print option.

Return a const iterator for the first print option.

**9.175.3.19 FirstTrain()**

```
TrainNumberMap::const_iterator TTSupport::TimeTableSystem::FirstTrain ( ) const [inline]
```

First train.

Return a const iterator for the first train.

**9.175.3.20 GetPrintOption()**

```
const char* TTSupport::TimeTableSystem::GetPrintOption (
    const char * key ) const [inline]
```

Fetch a print option.

Returns the value of a specified print option or the empty string if the print option was not found.

**Parameters**

<i>key</i>	The name of the print option.
------------	-------------------------------

#### 9.175.3.21 IthStation()

```
TTSupport::Station* TTSupport::TimeTableSystem::IthStation (
    int i ) [inline]
```

Return Ith station object.

Returns the NULL pointer if the index is out of range.

##### Parameters

<i>i</i>	The index of the station.
----------	---------------------------

References [i](#).

#### 9.175.3.22 LastCab()

```
CabNameMap::const_iterator TTSupport::TimeTableSystem::LastCab ( ) const [inline]
```

Last cab.

Return a const iterator for the last cab.

#### 9.175.3.23 LastPrintOption()

```
OptionHashMap::const_iterator TTSupport::TimeTableSystem::LastPrintOption ( ) const [inline]
```

Last Print option.

Return a const iterator for the last print option.

References [linuxgpio::in](#), and [linuxgpio::out](#).

#### 9.175.3.24 LastTrain()

```
TrainNumberMap::const_iterator TTSupport::TimeTableSystem::LastTrain ( ) const [inline]
```

Last train.

Return a const iterator for the last train.



### 9.175.3.25 MakeTimeTableGroupByClass()

```
bool TTSupport::TimeTableSystem::MakeTimeTableGroupByClass (
    ostream & out,
    TrainList & allTrains,
    TrainList & forwardTrains,
    TrainList & backwardTrains,
    char ** outmessage = NULL ) [private]
```

Make a time table grouped by class.

Writes a time table LaTeX file grouped by train class. Each class will have its own table in its own section. Returns true if successfull and false if there were problems (errors).

## Parameters

<i>out</i>	The LaTeX output stream.
<i>allTrains</i>	A list of all of the trains.
<i>forwardTrains</i>	A list of all forward moving trains.
<i>backwardTrains</i>	A list of all backward moving trains.
<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur.

## 9.175.3.26 MakeTimeTableGroupManually()

```
bool TTSupport::TimeTableSystem::MakeTimeTableGroupManually (
    ostream & out,
    int maxTrains,
    TrainList & allTrains,
    TrainList & forwardTrains,
    TrainList & backwardTrains,
    char ** outmessage = NULL ) [private]
```

Make a time table grouped manually.

Writes a time table LaTeX file grouped manually. Returns true if successfull and false if there were problems (errors).

## Parameters

<i>out</i>	The LaTeX output stream.
<i>maxTrains</i>	The maximum number of trains per table.
<i>allTrains</i>	A list of all of the trains.
<i>forwardTrains</i>	A list of all forward moving trains.
<i>backwardTrains</i>	A list of all backward moving trains.
<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur.

## 9.175.3.27 MakeTimeTableOneTable()

```
bool TTSupport::TimeTableSystem::MakeTimeTableOneTable (
    ostream & out,
    TrainList & allTrains,
    TrainList & forwardTrains,
    TrainList & backwardTrains,
    string header,
    string sectionTOP,
    char ** outmessage = NULL ) [private]
```

Make a time table as a single table.

Writes a time table LaTeX file with all trains in a single table. This only makes sense if the total number of trains is small.

## Parameters

<i>out</i>	The LaTeX output stream.
<i>allTrains</i>	A list of all of the trains.
<i>forwardTrains</i>	A list of all forward moving trains.
<i>backwardTrains</i>	A list of all backward moving trains.
<i>header</i>	String (LaTeX code) to use for the time table header.
<i>sectionTOP</i>	String (LaTeX code) to use for the section start.
<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur.

## 9.175.3.28 MakeTimeTableOneTableStationsCenter()

```
bool TTSupport::TimeTableSystem::MakeTimeTableOneTableStationsCenter (
    ostream & out,
    TrainList & forwardTrains,
    TrainList & backwardTrains,
    string header,
    string sectionTOP,
    char ** outmessage = NULL ) [private]
```

Make a time table as a single table, with the stations in the center (bi-directional trains).

## Parameters

<i>out</i>	The LaTeX output stream.
<i>forwardTrains</i>	A list of all forward moving trains.
<i>backwardTrains</i>	A list of all backward moving trains.
<i>header</i>	String (LaTeX code) to use for the time table header.
<i>sectionTOP</i>	String (LaTeX code) to use for the section start.
<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur.

## 9.175.3.29 MakeTimeTableOneTableStationsLeft()

```
bool TTSupport::TimeTableSystem::MakeTimeTableOneTableStationsLeft (
    ostream & out,
    TrainList & trains,
    string header,
    string sectionTOP,
    char ** outmessage = NULL ) [private]
```

Make a time table as a single table, with the stations on the left (single direction trains).

**Parameters**

<i>out</i>	The LaTeX output stream.
<i>trains</i>	A list of the trains.
<i>header</i>	String (LaTeX code) to use for the time table header.
<i>sectionTOP</i>	String (LaTeX code) to use for the section start.
<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur.

**9.175.3.30 Name()**

```
const char* TTSupport::TimeTableSystem::Name ( ) const [inline]
```

Return the name of the system.

**9.175.3.31 Note()**

```
const char* TTSupport::TimeTableSystem::Note (
    int i ) [inline]
```

Return the ith note (1-based!) as a string.

Returns the NULL pointer if the index is out of range.

**Parameters**

<i>i</i>	The note index. The first note is at index 1, not 0!.
----------	---

**9.175.3.32 NumberOfCabs()**

```
int TTSupport::TimeTableSystem::NumberOfCabs ( ) const [inline]
```

The number of cabs.

**9.175.3.33 NumberOfNotes()**

```
int TTSupport::TimeTableSystem::NumberOfNotes ( ) const [inline]
```

Return the number of notes.

#### 9.175.3.34 NumberOfStations()

```
int TTSupport::TimeTableSystem::NumberOfStations ( ) const [inline]
```

Number of stations.

Returns the number of stations in the system.

#### 9.175.3.35 NumberOfTrains()

```
int TTSupport::TimeTableSystem::NumberOfTrains ( ) const [inline]
```

Return the number of trains.

#### 9.175.3.36 ReadNote()

```
string TTSupport::TimeTableSystem::ReadNote (
    istream & in ) const [private]
```

Read in a note.

Reads a note from a stream. Returns the note text.

##### Parameters

<i>in</i>	Stream to read from.
-----------	----------------------

#### 9.175.3.37 SetDuplicateStationIndex()

```
void TTSupport::TimeTableSystem::SetDuplicateStationIndex (
    int i,
    int dup ) [inline]
```

Set the duplicate station index for a given station.

Only meaningful for out and back type layouts or layouts that have shared trackage. This would be stations along shared trackage. setting the duplicate station index indicates there is no duplicate station

##### Parameters

<i>i</i>	The index of the station to be updated.
<i>dup</i>	The other station index sharing this station location.

References i.

#### 9.175.3.38 SetNote()

```
bool TTSupport::TimeTableSystem::SetNote (
    int i,
    string note ) [inline]
```

Set the ith note (1-based!).

Updates the text of the specified note. Returns true if the note was updated or false if the index was out of range.

##### Parameters

<i>i</i>	The note index. The first note is at index 1, not 0!.
<i>note</i>	The new text for the note.

#### 9.175.3.39 SetPrintOption()

```
void TTSupport::TimeTableSystem::SetPrintOption (
    const char * key,
    string value ) [inline]
```

Set a print option.

Sets the value of a print option. Creates a new hash table element if the specified print option does not already exist.

##### Parameters

<i>key</i>	The name of the print option to be set.
<i>value</i>	The value to set the print option to.

#### 9.175.3.40 SMile()

```
double TTSupport::TimeTableSystem::SMile (
    int i ) const [inline]
```

Return the lth station's scale mile location.

Returns -1.0 if the index is out of range.

**Parameters**

<i>i</i>	The index of the station.
----------	---------------------------

References *i*.

**9.175.3.41 StationName()**

```
const char* TTSupport::TimeTableSystem::StationName (  
    int i ) const [inline]
```

Return the *i*th station name.

Returns the NULL pointer if the index is out of range.

**Parameters**

<i>i</i>	The index of the station.
----------	---------------------------

References *i*.

**9.175.3.42 TimeInterval()**

```
int TTSupport::TimeTableSystem::TimeInterval ( ) const [inline]
```

Return time interval.

**9.175.3.43 TimeScale()**

```
int TTSupport::TimeTableSystem::TimeScale ( ) const [inline]
```

Return time scale.

**9.175.3.44 TotalLength()**

```
double TTSupport::TimeTableSystem::TotalLength ( ) const [inline]
```

The total length of the route in scale miles.

This is just the scale mile location of the last station along the route.



#### 9.175.3.45 WriteNewTimeTableFile()

```
bool TTSupport::TimeTableSystem::WriteNewTimeTableFile (
    string filename = "TimeTableFile.tt",
    bool setfilename = false,
    char ** outmessage = NULL )
```

Write out a Time Table System to a new file.

The current contents of the time table is written to a new time table file. Returns true if successful and false if not.

##### Parameters

<i>filename</i>	The name of the file to write (if empty, use existing name, if available).
<i>setfilename</i>	Change the filename if true.
<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur.

#### 9.175.3.46 WriteNote()

```
ostream& TTSupport::TimeTableSystem::WriteNote (
    ostream & out,
    string note ) const [private]
```

Write out a note.

Writes the note text to a stream. Returns the stream.

##### Parameters

<i>out</i>	Stream to write to.
<i>note</i>	The note text.

#### 9.175.3.47 WriteOldTimeTableFile()

```
bool TTSupport::TimeTableSystem::WriteOldTimeTableFile (
    char ** outmessage = NULL ) [inline]
```

Write an old time table file.

The current contents of the time table is written to the file name stored in the time table object. This method just calls the WriteNewTimeTableFile method with the old file name. Returns true if successful and false if not.

**Parameters**

<i>outmessage</i>	Pointer to a pointer to receive any error messages for any errors that might occur.
-------------------	---

**9.175.4 Member Data Documentation****9.175.4.1 cabs**

`CabNameMap` `TTSupport::TimeTableSystem::cabs` [private]

Cap name map.

**9.175.4.2 DirectionName**

`string` `TTSupport::TimeTableSystem::DirectionName` [private]

Direction Name.

Used by print functions.

**9.175.4.3 filepath**

`PathName` `TTSupport::TimeTableSystem::filepath` [private]

The pathname of the file the system was loaded from.

**9.175.4.4 name**

`string` `TTSupport::TimeTableSystem::name` [private]

The name of the time table system.

#### 9.175.4.5 notes

```
vector<string> TTSupport::TimeTableSystem::notes [private]
```

Notes.

#### 9.175.4.6 printOptions

```
OptionHashMap TTSupport::TimeTableSystem::printOptions [private]
```

Print option hash table.

#### 9.175.4.7 stations

```
StationVector TTSupport::TimeTableSystem::stations [private]
```

Station stop vector.

#### 9.175.4.8 timeinterval

```
int TTSupport::TimeTableSystem::timeinterval [private]
```

Time interval.

#### 9.175.4.9 timescale

```
int TTSupport::TimeTableSystem::timescale [private]
```

Time scale.

#### 9.175.4.10 TOCP

```
bool TTSupport::TimeTableSystem::TOCP [private]
```

Table Of Contents?

Used by print functions.

## 9.175.4.11 trains

`TrainNumberMap` TTSupport::TimeTableSystem::trains [private]

`Train` number/symbol map.

## 9.176 CTCPanel::Toggle Class Reference

Toggle switch object type.

### Public Member Functions

- `Toggle` (name, \_ctcpanel, \_canvas,...)  
*Construct a toggle switch object.*
- `~Toggle` ()  
*Clean up all data objects and free up all resources.*
- `getv` ()  
*Method to get our value (lever position).*
- `setv` (state)  
*Method to set out value (level position).*
- `geti` (ind)  
*Method to get the state of one of our indicators (none).*
- `seti` (ind, value)  
*Method to set an indicator's state (none).*
- `invoke` ()  
*Method to invoke the toggle switch.*

### Private Member Functions

- `_VerifyOrientationHV` (option, value)  
*Method to validate an orientation option of horizontal or vertical.*
- `_configureLeftLabel` (option, value)  
*Method to update the leftlabel option.*
- `_configureRightLabel` (option, value)  
*Method to update the rightlabel option.*
- `_configureCenterLabel` (option, value)  
*Method to update the centerlabel option.*
- `_AddTLever` (pos)  
*Method to add (draw) a toggle switch lever.*
- `_MoveTLever` (mx, my)  
*Method to move an object's lever.*

## Private Attributes

- [ctcpanel](#)  
*The CTC Panel component (parent widget).*
- [canvas](#)  
*The canvas component (parent widget component).*
- [lever](#)

### 9.176.1 Detailed Description

Toggle switch object type.

These are on the control panel and represent simple toggle switches.

#### Parameters

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The control panel canvas to draw the switch plate on.
<code>...</code>	Options: <ul style="list-style-type: none"> <li>• <code>-x</code> The x coordinate of the object (readonly, default 0).</li> <li>• <code>-y</code> The y coordinate of the object (readonly, default 0).</li> <li>• <code>-controlpoint</code> The name of the control point this switch is part of (readonly, default CP1).</li> <li>• <code>-orientation</code> The orientation of the switch, either horizontal or vertical (readonly, default horizontal).</li> <li>• <code>-leftlabel</code> The label of the left or upper position (default "on").</li> <li>• <code>-rightlabel</code> The label of the right or lower position (default "off").</li> <li>• <code>-centerlabel</code> The label of the center position (default "off").</li> <li>• <code>-hascenter</code> Flag indicating if there is a center position or not (readonly, default no).</li> <li>• <code>-leftcommand</code> Script to run when the switch is in its left or upper position (default {}).</li> <li>• <code>-rightcommand</code> Script to run when the switch is in its right or lower position (default {}).</li> <li>• <code>-centercommand</code> Script to run when the switch is in its center position (default {}).</li> </ul>

Defined coords terminals:

- `xy` The base position of the object.

Defined values (states):

- `Left` Left position.
- `Center` Center position.
- `Right` Right position.

Defined indicators: none.

**Author**

Robert Heller <heller@deepsoft.com>

**9.176.2 Constructor & Destructor Documentation****9.176.2.1 Toggle()**

```
CTCPanel::Toggle::Toggle (
    name ,
    _ctcpanel ,
    _canvas ,
    ... )
```

Construct a toggle switch object.

**Parameters**

<code>_ctcpanel</code>	The <a href="#">CTCPanel</a> megawidget.
<code>_canvas</code>	The control panel canvas to draw the Code Button on.
<code>...</code>	Option list.

**9.176.2.2 ~Toggle()**

```
CTCPanel::Toggle::~~Toggle ( )
```

Clean up all data objects and free up all resources.

**9.176.3 Member Function Documentation****9.176.3.1 \_AddTLever()**

```
CTCPanel::Toggle::_AddTLever (
    pos ) [private]
```

Method to add (draw) a toggle switch lever.

## Parameters

<i>pos</i>	The lever's position (Left, Right, or Center).
------------	--

9.176.3.2 `_configureCenterLabel()`

```
CTCPanel::Toggle::_configureCenterLabel (
    option ,
    value ) [private]
```

Method to update the centerlabel option.

9.176.3.3 `_configureLeftLabel()`

```
CTCPanel::Toggle::_configureLeftLabel (
    option ,
    value ) [private]
```

Method to update the leftlabel option.

9.176.3.4 `_configureRightLabel()`

```
CTCPanel::Toggle::_configureRightLabel (
    option ,
    value ) [private]
```

Method to update the rightlabel option.

9.176.3.5 `_MoveTLever()`

```
CTCPanel::Toggle::_MoveTLever (
    mx ,
    my ) [private]
```

Method to move an object's lever.

**Parameters**

<i>mx</i>	Mouse X coordinate. The lever is moved to be near the mouse pointer.
<i>my</i>	Mouse Y coordinate. The lever is moved to be near the mouse pointer.

**9.176.3.6 \_VerifyOrientationHV()**

```
CTCPanel::Toggle::_VerifyOrientationHV (
    option ,
    value ) [private]
```

Method to validate an orientation option of horizontal or vertical.

**9.176.3.7 geti()**

```
CTCPanel::Toggle::geti (
    ind )
```

Method to get the state of one of our indicators (none).

**9.176.3.8 getv()**

```
CTCPanel::Toggle::getv ( )
```

Method to get our value (lever position).

**9.176.3.9 invoke()**

```
CTCPanel::Toggle::invoke ( )
```

Method to invoke the toggle switch.

One of the command scripts is executed depending on the lever position.



**9.176.3.10 seti()**

```
CTCPanel::Toggle::seti (
    ind ,
    value )
```

Method to set an indicator's state (none).

**9.176.3.11 setv()**

```
CTCPanel::Toggle::setv (
    state )
```

Method to set out value (level position).

**Parameters**

<i>state</i>	The new state to set.
--------------	-----------------------

**9.176.4 Member Data Documentation****9.176.4.1 canvas**

`CTCPanel::Toggle::canvas` [private]

The canvas component (parent widget component).

**9.176.4.2 ctcpnl**

`CTCPanel::Toggle::ctcpnl` [private]

The CTC Panel component (parent widget).

**9.176.4.3 lever**

`CTCPanel::Toggle::lever` [private]

**9.177 Parsers::TrackBody Class Reference**

List of track endpoints (T and E lines).

```
#include <TrackBody.h>
```

**Public Member Functions**

- [TrackBody](#) ([TrackBodyElt](#) \*Element, [TrackBody](#) \*Next)  
*Constructor.*
- [~TrackBody](#) ()  
*Destructor.*

## Static Public Member Functions

- static [TrackBody](#) \* [ConstTrackBody](#) ([TrackBodyElt](#) \*tbe, [TrackBody](#) \*tb)  
*Prepend a track endpoint.*
- static [TrackBody](#) \* [AppendTrackBodyElt](#) ([TrackBody](#) \*tb, [TrackBodyElt](#) \*tbe)  
*Append a track endpoint.*
- static int [TrackBodyLength](#) (const [TrackBody](#) \*tb)  
*Compute the count of track endpoints.*

## Private Attributes

- [TrackBodyElt](#) \* [element](#)  
*Current element.*
- [TrackBody](#) \* [next](#)  
*Next element.*

## Friends

- class [TrackGraph](#)
- ostream & [operator<<](#) (ostream &stream, [TrackBody](#) &track)  
*Output operator.*

### 9.177.1 Detailed Description

List of track endpoints (T and E lines).

#### Author

Robert Heller <heller@deepsoft.com>

### 9.177.2 Constructor & Destructor Documentation

#### 9.177.2.1 [TrackBody\(\)](#)

```
Parsers::TrackBody::TrackBody (  
    TrackBodyElt * Element,  
    TrackBody * Next ) [inline]
```

Constructor.

### 9.177.2.2 ~TrackBody()

```
Parsers::TrackBody::~~TrackBody ( ) [inline]
```

Destructor.

## 9.177.3 Member Function Documentation

### 9.177.3.1 AppendTrackBodyElt()

```
static TrackBody* Parsers::TrackBody::AppendTrackBodyElt (
    TrackBody * tb,
    TrackBodyElt * tbe ) [inline], [static]
```

Append a track endpoint.

References next.

### 9.177.3.2 ConsTrackBody()

```
static TrackBody* Parsers::TrackBody::ConsTrackBody (
    TrackBodyElt * tbe,
    TrackBody * tb ) [inline], [static]
```

Prepend a track endpoint.

### 9.177.3.3 TrackBodyLength()

```
static int Parsers::TrackBody::TrackBodyLength (
    const TrackBody * tb ) [inline], [static]
```

Compute the count of track endpoints.

References next.

## 9.177.4 Friends And Related Function Documentation

#### 9.177.4.1 operator<<

```
ostream& operator<< (
    ostream & stream,
    TrackBody & track ) [friend]
```

Output operator.

#### 9.177.4.2 TrackGraph

```
friend class TrackGraph [friend]
```

### 9.177.5 Member Data Documentation

#### 9.177.5.1 element

```
TrackBodyElt* Parsers::TrackBody::element [private]
```

Current element.

#### 9.177.5.2 next

```
TrackBody* Parsers::TrackBody::next [private]
```

Next element.

Referenced by AppendTrackBodyElt(), and TrackBodyLength().

## 9.178 Parsers::TrackBodyElt Class Reference

Track endpoint elements (T and E lines).

```
#include <TrackBody.h>
```

## Public Member Functions

- [TrackBodyElt](#) (int ind=-1, float X=-1.0, float Y=-1.0, float A=0.0)  
*Constructor.*
- [~TrackBodyElt](#) ()  
*Destructor.*

## Static Public Member Functions

- static [TrackBodyElt](#) \* [ConnectedTrackEnd](#) (int ind, float X, float Y, float A)  
*Create a connected track endpoint.*
- static [TrackBodyElt](#) \* [UnConnectedTrackEnd](#) (float X, float Y, float A)  
*Create a unconnected track endpoint.*

## Private Attributes

- int [index](#)  
*Index of connected track (T lines only).*
- float [x](#)  
*\$X\$ coordinate of track endpoint.*
- float [y](#)  
*\$Y\$ coordinate of track endpoint.*
- float [a](#)  
*Angle of track endpoint.*

## Friends

- class [TrackGraph](#)
- ostream & [operator<<](#) (ostream &stream, [TrackBodyElt](#) &elt)  
*Output operator.*

### 9.178.1 Detailed Description

Track endpoint elements (T and E lines).

#### Author

Robert Heller <heller@deepsoft.com>

### 9.178.2 Constructor & Destructor Documentation

### 9.178.2.1 TrackBodyElt()

```
Parsers::TrackBodyElt::TrackBodyElt (
    int ind = -1,
    float X = -1.0,
    float Y = -1.0,
    float A = 0.0 ) [inline]
```

Constructor.

### 9.178.2.2 ~TrackBodyElt()

```
Parsers::TrackBodyElt::~~TrackBodyElt ( ) [inline]
```

Destructor.

## 9.178.3 Member Function Documentation

### 9.178.3.1 ConnectedTrackEnd()

```
static TrackBodyElt* Parsers::TrackBodyElt::ConnectedTrackEnd (
    int ind,
    float X,
    float Y,
    float A ) [inline], [static]
```

Create a connected track endpoint.

References a, index, x, and y.

### 9.178.3.2 UnConnectedTrackEnd()

```
static TrackBodyElt* Parsers::TrackBodyElt::UnConnectedTrackEnd (
    float X,
    float Y,
    float A ) [inline], [static]
```

Create a unconnected track endpoint.

References a, x, and y.

## 9.178.4 Friends And Related Function Documentation

### 9.178.4.1 `operator<<`

```
ostream& operator<< (  
    ostream & stream,  
    TrackBodyElt & elt ) [friend]
```

Output operator.

### 9.178.4.2 `TrackGraph`

```
friend class TrackGraph [friend]
```

## 9.178.5 Member Data Documentation

### 9.178.5.1 `a`

```
float Parsers::TrackBodyElt::a [private]
```

Angle of track endpoint.

Referenced by `ConnectedTrackEnd()`, and `UnConnectedTrackEnd()`.

### 9.178.5.2 `index`

```
int Parsers::TrackBodyElt::index [private]
```

Index of connected track (T lines only).

Referenced by `ConnectedTrackEnd()`.



9.178.5.3 `x`

```
float Parsers::TrackBodyElt::x [private]
```

\$X\$ coordinate of track endpoint.

Referenced by `ConnectedTrackEnd()`, and `UnConnectedTrackEnd()`.

9.178.5.4 `y`

```
float Parsers::TrackBodyElt::y [private]
```

\$Y\$ coordinate of track endpoint.

Referenced by `ConnectedTrackEnd()`, and `UnConnectedTrackEnd()`.

## 9.179 Parsers::TrackGraph Class Reference

Track Graph class, which encapsulates the track graphs.

```
#include <TrackGraph.h>
```

### Classes

- struct [CompressedEdgeValues](#)  
*Compressed graph edge values.*
- struct [CompressedNodeValues](#)  
*Compressed graph node values.*
- struct [EdgeValues](#)  
*Uncompressed graph edge values.*
- struct [NodeValues](#)  
*Uncompressed graph node values.*
- struct [Point](#)  
*Position structure.*
- class [Transform2D](#)  
*Two dimensional transform class.*

### Public Types

- enum [NodeType](#) {  
  [Undefined](#) = -1, [Track](#) = 0, [Turnout](#), [Turntable](#),  
  [Block](#), [SwitchMotor](#), [Signal](#), [Sensor](#),  
  [Control](#) }  
*Node types.*
- typedef std::pair< int, int > [CompressedEdgePair](#)
- typedef std::vector< [CompressedEdgePair](#) > [CompressedEdgePairVector](#)

## Public Member Functions

- [TrackGraph](#) ()  
*Constructor.*
- [~TrackGraph](#) ()  
*Destructor.*
- void [InsertCurveTrack](#) (int number, [TrackBody](#) \*tb, float orgX, float orgY, float radius)  
*Insert a (circular) curved piece of track.*
- void [InsertBezierTrack](#) (int number, [BezierBody](#) \*trb, float x1, float y1, float x2, float y2, float x3, float y3, float x4, float y4)  
*Insert a bezier curved piece of track.*
- void [InsertCornuTrack](#) (int number, [CornuBody](#) \*trb, float pos1x, float pos1y, float angle1, float radius1, float center1x, float center1y, float pos2x, float pos2y, float angle2, float radius2, float center2x, float center2y)  
*Insert a cornu curved piece of track.*
- void [InsertStraightTrack](#) (int number, [TrackBody](#) \*tb)  
*Insert a straight piece of track.*
- void [InsertJointTrack](#) (int number, [TrackBody](#) \*tb, float l0, float l1, float [angle](#), float R, float L)  
*Insert a (spiral) curved piece of track.*
- void [InsertTurnOut](#) (int number, float orgX, float orgY, float orient, const char \*name, [TurnoutBody](#) \*trb)  
*Insert a turnout or crossing.*
- void [InsertTurnTable](#) (int number, float orgX, float orgY, float radius, [TrackBody](#) \*tb)  
*Insert a turntable.*
- void [InsertBlock](#) (int number, char \*\_name, char \*\_script, [IntegerList](#) \*\_tracklist)  
*Insert a Block.*
- void [InsertSwitchMotor](#) (int number, int turnout, char \*\_name, char \*\_normal, char \*\_reverse, char \*\_pointsense)  
*Insert a switch motor.*
- void [InsertSignal](#) (int number, char \*\_name, float \_origx, float \_origy, float \_angle, int \_numheads, StringPairList \*\_aspects)  
*Insert a signal.*
- void [InsertSensor](#) (int number, char \*\_name, float \_origx, float \_origy, char \*\_sensescript)  
*Insert a sensor.*
- void [InsertControl](#) (int number, char \*\_name, float \_origx, float \_origy, char \*\_onscript, char \*\_offscript)  
*Insert a control.*
- bool [IsNodeP](#) (int nid) const  
*Tests if a node id exists in the graph.*
- int [NumEdges](#) (int nid) const  
*Returns the number of edges for the specified node id.*
- int [EdgeIndex](#) (int nid, int edgenum) const  
*Returns the node id of the specified edge of the node.*
- float [EdgeX](#) (int nid, int edgenum) const  
*Returns the \$X\$ coordinate of the specified edge of the node.*
- float [EdgeY](#) (int nid, int edgenum) const  
*Returns the \$Y\$ coordinate of the specified edge of the node.*
- float [EdgeA](#) (int nid, int edgenum) const  
*Returns the angle of the specified edge of the node.*
- float [EdgeLength](#) (int nid, int edgenum) const  
*Returns the length of an edge.*
- [NodeType](#) [TypeOfNode](#) (int nid) const

- Returns the type of the node.*

  - const [TurnoutGraphic](#) \* [NodeTurnoutGraphic](#) (int nid) const
- Returns the [TurnoutGraphic](#) of the node.*

  - const [TurnoutRoutelist](#) \* [NodeTurnoutRoutelist](#) (int nid) const
- Returns the [TurnoutRoutelist](#) of the node.*

  - float [LengthOfNode](#) (int nid) const
- Return the track length of a node.*

  - const [IntegerList](#) \* [TrackList](#) (int nid) const
- Return a block's tracklist.*

  - int [TurnoutNumber](#) (int nid) const
- Return a switchmotor's turnout number.*

  - const char \* [NameOfNode](#) (int nid) const
- Return a block's or switchmotor's name.*

  - const char \* [SenseScript](#) (int nid) const
- Return a block's or switchmotor's sense script.*

  - const char \* [NormalActionScript](#) (int nid) const
- Return a switchmotor's normal action script.*

  - const char \* [ReverseActionScript](#) (int nid) const
- Return a block's or switchmotor's reverse action script.*

  - int [NumberOfHeads](#) (int nid) const
- Return a Signal's number of heads.*

  - const StringPairList \* [SignalAspects](#) (int nid) const
- Return a Signal's aspect list.*

  - float [OrigX](#) (int nid) const
- Return the  $x$  coordinate of the signal base.*

  - float [OrigY](#) (int nid) const
- Return the  $y$  coordinate of the signal base.*

  - float [Angle](#) (int nid) const
- Return the angle of the signal.*

  - const char \* [OnScript](#) (int nid) const
- Return the on action script.*

  - const char \* [OffScript](#) (int nid) const
- Return the off action script.*

  - int [LowestNode](#) () const
- Returns the lowest numbered node id.*

  - int [HighestNode](#) () const
- Returns the highest numbered node id.*

  - void [CompressGraph](#) ()
- Create a compressed graph.*

  - bool [IsCompressedNode](#) (int cnid) const
- Is cid a node in the compressed graph?*

  - int [CompressedEdgeCount](#) (int cnid) const
- Number of compressed graph edges for node cnid.*

  - float [CompressedEdgeLength](#) (int cnid, int edgenum) const
- Length of a compressed graph edge.*

  - int [CompressedEdgeNode](#) (int cnid, int edgenum) const
- Next Edge node.*

- [IntegerList](#) \* [CompressedNodeSegments](#) (int cnid) const  
*Raw nodes in a compressed graph node.*
- double [CompressedNodePositionX](#) (int cnid) const  
*X Coordinate of a Compressed Node position.*
- double [CompressedNodePositionY](#) (int cnid) const  
*X Coordinate of a Compressed Node position.*
- bool [IsCompressed](#) () const  
*Is the graph compressed?*
- const [IntegerList](#) \* [Heads](#) ()  
*Uncompressed graph heads.*
- const [IntegerList](#) \* [Roots](#) ()  
*Compressed graph roots.*
- void [CompressedGraphCircleLayout](#) (double radius)  
*Run the BGL circle\_graph\_layout for a given radius.*
- bool [CompressedGraphKamadaKawaiSpring](#) (double sidelength)  
*Run the BGL kamada\_kawai\_spring\_layout for a given side length.*
- [CompressedEdgePairVector](#) [CompressedGraphKruskalMinimumSpanningTree](#) ()  
*Run the kruskal\_minimum\_spanning\_tree algorithm and return a vector of edge pairs.*
- [CompressedEdgePairVector](#) [CompressedGraphPrimMinimumSpanningTree](#) ()  
*Run the prim\_minimum\_spanning\_tree algorithm and return a Parent Vector.*

### Static Public Member Functions

- static float [LengthOfStraight](#) (float x1, float y1, float x2, float y2)  
*Compute the length of a piece of straight track.*
- static float [LengthOfCurve](#) (float radius, float a1, float a2)  
*Compute the length of a (circular) curved piece of track.*
- static float [LengthOfJoint](#) (float l0, float l1, float [angle](#), float R, float L)  
*Compute the length of a (spiral) curved piece of track.*

### Private Types

- enum [RotationUnit](#) { [Degrees](#), [Radians](#) }  
*Rotational units.*
- typedef adjacency\_list< vecS, vecS, directedS, [NodeValues](#), [EdgeValues](#) > [Graph](#)  
*Boost Graph type (adjacency\_list).*
- typedef graph\_traits< [Graph](#) >::vertex\_descriptor [Node](#)  
*Vertex type.*
- typedef std::map< int, [Node](#) > [IdNodeMap](#)  
*Type of Node Id map.*
- typedef adjacency\_list< vecS, vecS, undirectedS, [CompressedNodeValues](#), [CompressedEdgeValues](#) > [CompressedGraph](#)  
*Boost Compressed Graph type (adjacency\_list).*
- typedef graph\_traits< [CompressedGraph](#) >::vertex\_descriptor [CompressedNode](#)  
*Compressed Graph Vertex type.*
- typedef std::map< int, [CompressedNode](#) > [CompressedIdNodeMap](#)  
*Type of Node Id map.*
- typedef std::vector< graph\_traits< [CompressedGraph](#) >::vertex\_descriptor > [CompressedNodeVector](#)

## Private Member Functions

- [Node AddNewNode](#) (int id, [NodeType](#) \_type=[Undefined](#), [TurnoutGraphic](#) \*\_tgr=[NULL](#), [TurnoutRoutelist](#) \*\_tpo=[NULL](#), float \_length=0.0)
 

*Helper function to create a new node.*
- void [computeHeads](#) ()
 

*Compute uncompressed graph heads (calls strong\_components).*
- [CompressedNode insertCompressedNode](#) ([Node](#) rawnode)
 

*Insert a compressed graph node.*
- bool [IsNone](#) ([Node](#) node)
 

*Check if node is the none node;.*
- void [traversePrimMST](#) ([CompressedEdgePairVector](#) &result, [CompressedNodeVector](#) &parents, [CompressedNode](#) r) const
 

*Traverse a PrimMST, starting at root r, inserting EdgePairs into result.*
- [Node FindNode](#) (int index) const
 

*Find a node in the hash table.*
- [TurnoutGraphic](#) \* [MakeTurnoutGraphic](#) (float orgX, float orgY, float orient, [TurnoutBody](#) \*trb)
 

*Generate a turnout node's graphic.*
- [TurnoutRoutelist](#) \* [MakeTurnoutRouteList](#) ([TurnoutBody](#) \*trb, const [TurnoutGraphic](#) \*tgr, float &length)
 

*Generate a turnout node's route list.*
- bool [compressed\\_edge\\_exists](#) ([CompressedNode](#) cnode1, [CompressedNode](#) cnode2) const
 

*Test if an edge already exists.*
- [Node FindBlock](#) ([Node](#) node) const
 

*Find the block this (raw) nodeid is in.*
- [Transform2D](#) \* [tr\\_translate](#) (float x, float y)
 

*Construct a translation transform.*
- [Transform2D](#) \* [tr\\_scale](#) (float mag\_factor)
 

*Construct a uniform scale transform.*
- [Transform2D](#) \* [tr\\_scale](#) (float xscale, float yscale)
 

*Construct a non-uniform scale transform.*
- [Transform2D](#) \* [tr\\_rotate](#) (float amount, [RotationUnit](#) measure)
 

*Construct a rotational transform.*

## Static Private Member Functions

- static void [DeleteTurnoutGraphic](#) ([TurnoutGraphic](#) \*tgr)
 

*Free up the memory used by a turnout node's graphic.*
- static void [DeleteTurnoutRouteList](#) ([TurnoutRoutelist](#) \*tpo)
 

*Free up the memory used by a turnout node's route list.*
- static float [ComputeRouteLength](#) (const [TurnoutGraphic](#) \*tgr, const [IntegerList](#) \*il)
 

*Compute the length of a route.*

## Private Attributes

- [Graph nodes](#)  
*Graph adjacency\_list.*
- [IdNodeMap idMap](#)  
*Node Id map.*
- [IntegerList \\* heads](#)  
*Uncompressed graph heads (strong components).*
- [bool valid\\_heads](#)  
*Flag to indicate if heads is valid.*
- [Node none](#)  
*Special node that is nowhere (where all unconnected trackage goes).*
- [CompressedGraph c\\_nodes](#)  
*Compressed Graph adjacency\_list.*
- [CompressedIdNodeMap c\\_idMap](#)  
*Node Id map.*
- [IntegerList \\* c\\_roots](#)  
*Compressed Graph Roots.*
- [bool compressedP](#)  
*Is graph compressed?*
- [bool circleLayoutP](#)  
*Has CompressedGraphCircleLayout been run?*
- [bool KamadaKawaiSpringLayoutP](#)  
*Has CompressedGraphKamadaKawaiSpring been run?*
- [std::map< Node, CompressedNode > backpointers](#)  
*Backpointer map.*

## Friends

- [std::ostream & operator<<](#) (ostream &stream, [TrackGraph](#) &graph)  
*Output operator.*

### 9.179.1 Detailed Description

Track Graph class, which encapsulates the track graphs.

Holds the two track graphs, an uncompressed, directed graph built from the layout file and a compressed, undirected graph where successive segments of plain trackage are collapsed into a single node. Both graphs use the Boost Graph Library `adjacency_list` template class as the basic graph implementation class. All nodes in both graphs have a unique node id, which is the XTrkCad layout object number.

Several of the Boost Graph Library graph algorithms are implemented, including `circle_graph_layout()`, `kamada_kawai_↔_spring_layout()`, `kruskal_minimum_spanning_tree()` and `prim_minimum_spanning_tree()`. In addition, the `strong_↔_components()` algorithm is used to gather nodes into one or more connected groups, since sometimes model train layouts have disjoint sections of track. An example would be a regular main line and an isolated mining or logging railroad. Another example would be a regular main line and one (or more) mass transit (eg trolley) line(s). The head nodes of the collected groups are accessed with the member function [Heads\(\)](#), which returns a list of nodes that are

the heads of each of the connected groups. The [CompressGraph\(\)](#) function will compress each connected group into a separately compressed graph, with its own root node. The root nodes of each of the compressed sub-graphs are returned with the [Roots\(\)](#) member function.

The computation of connected groups and graph compression are implemented using a lazy eval methodology. The connected group collection process is not run until the member function [Heads\(\)](#) is called to actually access the list of group heads. And the graph compression is not called until the member function [Roots\(\)](#) is called. Calling the [CompressGraph\(\)](#) member function (or the [Roots\(\)](#) member function), will also call the connected group collection process if it is needed.

To help discover possible mainlines, two minimum spanning tree (MST) algorithms are provided, Kruskal's and Prim's, via the member functions [CompressedGraphKruskalMinimumSpanningTree\(\)](#) and [CompressedGraphPrimMinimumSpanningTree\(\)](#), respectively. Both of these function return a list of edge pairs: a STL vector of STL pairs of compressed nodes.

Two graphical layout algorithms are also provided, circle graph layout and kamada kawai spring layout, via the two member functions, [CompressedGraphCircleLayout\(\)](#) and [CompressedGraphKamadaKawaiSpring\(\)](#), respectively. Both set or update a pair of double values  $(x,y)$  associated with every compressed node and which are accessed with the [CompressedNodePositionX\(\)](#) and [CompressedNodePositionY\(\)](#) member functions. [CompressedGraphCircleLayout\(\)](#) sets these files and [CompressedGraphKamadaKawaiSpring\(\)](#) updates them. [CompressedGraphCircleLayout\(\)](#) should be called before [CompressedGraphKamadaKawaiSpring\(\)](#). [CompressedGraphKamadaKawaiSpring\(\)](#) returns a boolean flag indicating if it was able to create a layout. Returning a false value indicates that the graph was disjoint – [CompressedGraphKamadaKawaiSpring\(\)](#) cannot be used with a disjoint graph.

#### Author

Robert Heller <heller@deepsoft.com>

## 9.179.2 Member Typedef Documentation

### 9.179.2.1 CompressedEdgePair

```
typedef std::pair< int, int > Parsers::TrackGraph::CompressedEdgePair
```

### 9.179.2.2 CompressedEdgePairVector

```
typedef std::vector< CompressedEdgePair > Parsers::TrackGraph::CompressedEdgePairVector
```

### 9.179.2.3 CompressedGraph

```
typedef adjacency_list< vecS, vecS, undirectedS, CompressedNodeValues , CompressedEdgeValues >  
Parsers::TrackGraph::CompressedGraph [private]
```

Boost Compressed Graph type (adjacency\_list).

### 9.179.2.4 CompressedIdNodeMap

```
typedef std::map< int, CompressedNode > Parsers::TrackGraph::CompressedIdNodeMap [private]
```

Type of Node Id map.

### 9.179.2.5 CompressedNode

```
typedef graph_traits< CompressedGraph >::vertex_descriptor Parsers::TrackGraph::CompressedNode  
[private]
```

Compressed Graph Vertex type.

### 9.179.2.6 CompressedNodeVector

```
typedef std::vector< graph_traits < CompressedGraph >::vertex_descriptor > Parsers::TrackGraph↵  
::CompressedNodeVector [private]
```

### 9.179.2.7 Graph

```
typedef adjacency_list< vecS, vecS, directedS, NodeValues , EdgeValues > Parsers::TrackGraph::↵  
Graph [private]
```

Boost Graph type (adjacency\_list).



### 9.179.2.8 IdNodeMap

```
typedef std::map< int, Node > Parsers::TrackGraph::IdNodeMap [private]
```

Type of Node Id map.

### 9.179.2.9 Node

```
typedef graph_traits< Graph >::vertex_descriptor Parsers::TrackGraph::Node [private]
```

Vertex type.

## 9.179.3 Member Enumeration Documentation

### 9.179.3.1 NodeType

```
enum Parsers::TrackGraph::NodeType
```

Node types.

Enumerator

Undefined	Undefined type.
Track	Plain trackage: straight, curved, or easement.
Turnout	Turnout or crossing.
Turntable	Turntable.
Block	Block.
SwitchMotor	Switch Motor.
Signal	Signal.
Sensor	Sensor.
Control	Control.

### 9.179.3.2 RotationUnit

```
enum Parsers::TrackGraph::RotationUnit [private]
```

Rotational units.

**Enumerator**

Degrees	Units are in degrees.
Radians	Units are in radians.

**9.179.4 Constructor & Destructor Documentation****9.179.4.1 TrackGraph()**

```
Parsers::TrackGraph::TrackGraph ( )
```

Constructor.

**9.179.4.2 ~TrackGraph()**

```
Parsers::TrackGraph::~~TrackGraph ( )
```

Destructor.

**9.179.5 Member Function Documentation****9.179.5.1 AddNewNode()**

```
Node Parsers::TrackGraph::AddNewNode (
    int id,
    NodeType _type = Undefined,
    TurnoutGraphic * _tgr = NULL,
    TurnoutRoutelist * _tpo = NULL,
    float _length = 0.0 ) [private]
```

Helper function to create a new node.

**9.179.5.2 Angle()**

```
float Parsers::TrackGraph::Angle (
    int nid ) const
```

Return the angle of the signal.

## Parameters

<i>nid</i>	The node to look at.
------------	----------------------

Referenced by Parsers::LayoutFile::Angle().

### 9.179.5.3 compressed\_edge\_exists()

```
bool Parsers::TrackGraph::compressed_edge_exists (
    CompressedNode cnode1,
    CompressedNode cnode2 ) const [private]
```

Test if an edge already exists.

### 9.179.5.4 CompressedEdgeCount()

```
int Parsers::TrackGraph::CompressedEdgeCount (
    int cnid ) const
```

Number of compressed graph edges for node cnid.

Referenced by Parsers::LayoutFile::CompressedEdgeCount().

### 9.179.5.5 CompressedEdgeLength()

```
float Parsers::TrackGraph::CompressedEdgeLength (
    int cnid,
    int edgenum ) const
```

Length of a compressed graph edge.

Referenced by Parsers::LayoutFile::CompressedEdgeLength().

### 9.179.5.6 CompressedEdgeNode()

```
int Parsers::TrackGraph::CompressedEdgeNode (
    int cnid,
    int edgenum ) const
```

Next Edge node.

Referenced by Parsers::LayoutFile::CompressedEdgeNode().

#### 9.179.5.7 CompressedGraphCircleLayout()

```
void Parsers::TrackGraph::CompressedGraphCircleLayout (
    double radius )
```

Run the BGL circle\_graph\_layout for a given radius.

Referenced by Parsers::LayoutFile::CompressedGraphCircleLayout().

#### 9.179.5.8 CompressedGraphKamadaKawaiSpring()

```
bool Parsers::TrackGraph::CompressedGraphKamadaKawaiSpring (
    double sidelength )
```

Run the BGL kamada\_kawai\_spring\_layout for a given side length.

Referenced by Parsers::LayoutFile::CompressedGraphKamadaKawaiSpring().

#### 9.179.5.9 CompressedGraphKruskalMinimumSpanningTree()

```
CompressedEdgePairVector Parsers::TrackGraph::CompressedGraphKruskalMinimumSpanningTree ( )
```

Run the kruskal\_minimum\_spanning\_tree algorithm and return a vector of edge pairs.

Referenced by Parsers::LayoutFile::CompressedGraphKruskalMinimumSpanningTree().

#### 9.179.5.10 CompressedGraphPrimMinimumSpanningTree()

```
CompressedEdgePairVector Parsers::TrackGraph::CompressedGraphPrimMinimumSpanningTree ( )
```

Run the prim\_minimum\_spanning\_tree algorithm and return a Parent Vector.

Referenced by Parsers::LayoutFile::CompressedGraphPrimMinimumSpanningTree().

**9.179.5.11 CompressedNodePositionX()**

```
double Parsers::TrackGraph::CompressedNodePositionX (
    int cnid ) const
```

X Coordinate of a Compressed Node position.

Referenced by Parsers::LayoutFile::CompressedNodePositionX().

**9.179.5.12 CompressedNodePositionY()**

```
double Parsers::TrackGraph::CompressedNodePositionY (
    int cnid ) const
```

X Coordinate of a Compressed Node position.

Referenced by Parsers::LayoutFile::CompressedNodePositionY().

**9.179.5.13 CompressedNodeSegments()**

```
IntegerList* Parsers::TrackGraph::CompressedNodeSegments (
    int cnid ) const
```

Raw nodes in a compressed graph node.

Referenced by Parsers::LayoutFile::CompressedNodeSegments().

**9.179.5.14 CompressGraph()**

```
void Parsers::TrackGraph::CompressGraph ( )
```

Create a compressed graph.

Referenced by Parsers::LayoutFile::CompressGraph().

**9.179.5.15 computeHeads()**

```
void Parsers::TrackGraph::computeHeads ( ) [private]
```

Compute uncompressed graph heads (calls strong\_components).

**9.179.5.16 ComputeRouteLength()**

```
static float Parsers::TrackGraph::ComputeRouteLength (
    const TurnoutGraphic * tgr,
    const IntegerList * il ) [static], [private]
```

Compute the length of a route.

**9.179.5.17 DeleteTurnoutGraphic()**

```
static void Parsers::TrackGraph::DeleteTurnoutGraphic (
    TurnoutGraphic * tgr ) [static], [private]
```

Free up the memory used by a turnout node's graphic.

**9.179.5.18 DeleteTurnoutRouteList()**

```
static void Parsers::TrackGraph::DeleteTurnoutRouteList (
    TurnoutRoutelist * tpo ) [static], [private]
```

Free up the memory used by a turnout node's route list.

**9.179.5.19 EdgeA()**

```
float Parsers::TrackGraph::EdgeA (
    int nid,
    int edgenum ) const
```

Returns the angle of the specified edge of the node.

Referenced by Parsers::LayoutFile::EdgeA().

**9.179.5.20 EdgeIndex()**

```
int Parsers::TrackGraph::EdgeIndex (
    int nid,
    int edgenum ) const
```

Returns the node id of the specified edge of the node.

Referenced by Parsers::LayoutFile::EdgeIndex().

#### 9.179.5.21 EdgeLength()

```
float Parsers::TrackGraph::EdgeLength (
    int nid,
    int edgenum ) const
```

Returns the length of an edge.

Referenced by Parsers::LayoutFile::EdgeLength().

#### 9.179.5.22 EdgeX()

```
float Parsers::TrackGraph::EdgeX (
    int nid,
    int edgenum ) const
```

Returns the  $XX$  coordinate of the specified edge of the node.

Referenced by Parsers::LayoutFile::EdgeX().

#### 9.179.5.23 EdgeY()

```
float Parsers::TrackGraph::EdgeY (
    int nid,
    int edgenum ) const
```

Returns the  $YY$  coordinate of the specified edge of the node.

Referenced by Parsers::LayoutFile::EdgeY().

#### 9.179.5.24 FindBlock()

```
Node Parsers::TrackGraph::FindBlock (
    Node node ) const [private]
```

Find the block this (raw) nodeid is in.

##### Parameters

<i>node</i>	The (raw) Node.
-------------	-----------------

**Returns**

the raw Node of the block node or none, if there is no block.

**9.179.5.25 FindNode()**

```
Node Parsers::TrackGraph::FindNode (
    int index ) const [private]
```

Find a node in the hash table.

**9.179.5.26 Heads()**

```
const IntegerList* Parsers::TrackGraph::Heads ( ) [inline]
```

Uncompressed graph heads.

Referenced by Parsers::LayoutFile::Heads().

**9.179.5.27 HighestNode()**

```
int Parsers::TrackGraph::HighestNode ( ) const
```

Returns the highest numbered node id.

Referenced by Parsers::LayoutFile::HighestNode().

**9.179.5.28 InsertBezierTrack()**

```
void Parsers::TrackGraph::InsertBezierTrack (
    int number,
    BezierBody * trb,
    float x1,
    float y1,
    float x2,
    float y2,
    float x3,
    float y3,
    float x4,
    float y4 )
```

Insert a bezier curved piece of track.



#### 9.179.5.29 InsertBlock()

```
void Parsers::TrackGraph::InsertBlock (
    int number,
    char * _name,
    char * _script,
    IntegerList * _tracklist )
```

Insert a Block.

#### 9.179.5.30 insertCompressedNode()

```
CompressedNode Parsers::TrackGraph::insertCompressedNode (
    Node rawnode ) [private]
```

Insert a compressed graph node.

#### 9.179.5.31 InsertControl()

```
void Parsers::TrackGraph::InsertControl (
    int number,
    char * _name,
    float _origx,
    float _origy,
    char * _onscript,
    char * _offscript )
```

Insert a control.

#### 9.179.5.32 InsertCornuTrack()

```
void Parsers::TrackGraph::InsertCornuTrack (
    int number,
    CornuBody * trb,
    float pos1x,
    float pos1y,
    float angle1,
    float radius1,
    float center1x,
    float center1y,
    float pos2x,
    float pos2y,
    float angle2,
    float radius2,
    float center2x,
    float center2y )
```

Insert a cornu curved piece of track.

#### 9.179.5.33 InsertCurveTrack()

```
void Parsers::TrackGraph::InsertCurveTrack (
    int number,
    TrackBody * tb,
    float orgX,
    float orgY,
    float radius )
```

Insert a (circular) curved piece of track.

#### 9.179.5.34 InsertJointTrack()

```
void Parsers::TrackGraph::InsertJointTrack (
    int number,
    TrackBody * tb,
    float l0,
    float l1,
    float angle,
    float R,
    float L )
```

Insert a (spiral) curved piece of track.

#### 9.179.5.35 InsertSensor()

```
void Parsers::TrackGraph::InsertSensor (
    int number,
    char * _name,
    float _origx,
    float _origy,
    char * _sensescript )
```

Insert a sensor.

#### 9.179.5.36 InsertSignal()

```
void Parsers::TrackGraph::InsertSignal (
    int number,
    char * _name,
    float _origx,
    float _origy,
    float _angle,
    int _numheads,
    StringPairList * _aspects )
```

Insert a signal.

#### 9.179.5.37 InsertStraightTrack()

```
void Parsers::TrackGraph::InsertStraightTrack (
    int number,
    TrackBody * tb )
```

Insert a straight piece of track.

#### 9.179.5.38 InsertSwitchMotor()

```
void Parsers::TrackGraph::InsertSwitchMotor (
    int number,
    int turnout,
    char * _name,
    char * _normal,
    char * _reverse,
    char * _pointsense )
```

Insert a switch motor.

#### 9.179.5.39 InsertTurnOut()

```
void Parsers::TrackGraph::InsertTurnOut (
    int number,
    float orgX,
    float orgY,
    float orient,
    const char * name,
    TurnoutBody * trb )
```

Insert a turnout or crossing.

#### 9.179.5.40 InsertTurnTable()

```
void Parsers::TrackGraph::InsertTurnTable (
    int number,
    float orgX,
    float orgY,
    float radius,
    TrackBody * tb )
```

Insert a turntable.

#### 9.179.5.41 IsCompressed()

```
bool Parsers::TrackGraph::IsCompressed ( ) const [inline]
```

Is the graph compressed?

##### Returns

True if the graph is compressed, False otherwise.

Referenced by Parsers::LayoutFile::IsCompressed().

#### 9.179.5.42 IsCompressedNode()

```
bool Parsers::TrackGraph::IsCompressedNode (
    int cnid ) const
```

Is cid a node in the compressed graph?

Referenced by Parsers::LayoutFile::IsCompressedNode().

#### 9.179.5.43 IsNodeP()

```
bool Parsers::TrackGraph::IsNodeP (
    int nid ) const
```

Tests if a node id exists in the graph.

Referenced by Parsers::LayoutFile::IsNodeP().

#### 9.179.5.44 IsNone()

```
bool Parsers::TrackGraph::IsNone (
    Node node ) [inline], [private]
```

Check if node is the none node;.

#### 9.179.5.45 LengthOfCurve()

```
static float Parsers::TrackGraph::LengthOfCurve (
    float radius,
    float a1,
    float a2 ) [static]
```

Compute the length of a (circular) curved piece of track.

#### 9.179.5.46 LengthOfJoint()

```
static float Parsers::TrackGraph::LengthOfJoint (
    float l0,
    float l1,
    float angle,
    float R,
    float L ) [static]
```

Compute the length of a (spiral) curved piece of track.

#### 9.179.5.47 LengthOfNode()

```
float Parsers::TrackGraph::LengthOfNode (
    int nid ) const
```

Return the track length of a node.

Referenced by Parsers::LayoutFile::LengthOfNode().

#### 9.179.5.48 LengthOfStraight()

```
static float Parsers::TrackGraph::LengthOfStraight (
    float x1,
    float y1,
    float x2,
    float y2 ) [static]
```

Compute the length of a piece of straight track.

**9.179.5.49 LowestNode()**

```
int Parsers::TrackGraph::LowestNode ( ) const
```

Returns the lowest numbered node id.

Referenced by Parsers::LayoutFile::LowestNode().

**9.179.5.50 MakeTurnoutGraphic()**

```
TurnoutGraphic* Parsers::TrackGraph::MakeTurnoutGraphic (
    float orgX,
    float orgY,
    float orient,
    TurnoutBody * trb ) [private]
```

Generate a turnout node's graphic.

**9.179.5.51 MakeTurnoutRouteList()**

```
TurnoutRoutelist* Parsers::TrackGraph::MakeTurnoutRouteList (
    TurnoutBody * trb,
    const TurnoutGraphic * tgr,
    float & length ) [private]
```

Generate a turnout node's route list.

**9.179.5.52 NameOfNode()**

```
const char* Parsers::TrackGraph::NameOfNode (
    int nid ) const
```

Return a block's or switchmotor's name.

Referenced by Parsers::LayoutFile::NameOfNode().

**9.179.5.53 NodeTurnoutGraphic()**

```
const TurnoutGraphic* Parsers::TrackGraph::NodeTurnoutGraphic (
    int nid ) const
```

Returns the [TurnoutGraphic](#) of the node.

Referenced by [Parsers::LayoutFile::NodeTurnoutGraphic\(\)](#).

**9.179.5.54 NodeTurnoutRoutelist()**

```
const TurnoutRoutelist* Parsers::TrackGraph::NodeTurnoutRoutelist (
    int nid ) const
```

Returns the [TurnoutRoutelist](#) of the node.

Referenced by [Parsers::LayoutFile::NodeTurnoutRoutelist\(\)](#).

**9.179.5.55 NormalActionScript()**

```
const char* Parsers::TrackGraph::NormalActionScript (
    int nid ) const
```

Return a switchmotor's normal action script.

Referenced by [Parsers::LayoutFile::NormalActionScript\(\)](#).

**9.179.5.56 NumberOfHeads()**

```
int Parsers::TrackGraph::NumberOfHeads (
    int nid ) const
```

Return a Signal's number of heads.

Referenced by [Parsers::LayoutFile::NumberOfHeads\(\)](#).

**9.179.5.57 NumEdges()**

```
int Parsers::TrackGraph::NumEdges (
    int nid ) const
```

Returns the number of edges for the specified node id.

Referenced by [Parsers::LayoutFile::NumEdges\(\)](#).

**9.179.5.58 OffScript()**

```
const char* Parsers::TrackGraph::OffScript (
    int nid ) const
```

Return the off action script.

**Parameters**

<i>nid</i>	The node to look at.
------------	----------------------

Referenced by `Parsers::LayoutFile::OffScript()`.

**9.179.5.59 OnScript()**

```
const char* Parsers::TrackGraph::OnScript (  
    int nid ) const
```

Return the on action script.

**Parameters**

<i>nid</i>	The node to look at.
------------	----------------------

Referenced by `Parsers::LayoutFile::OnScript()`.

**9.179.5.60 OrigX()**

```
float Parsers::TrackGraph::OrigX (  
    int nid ) const
```

Return the  $x$  coordinate of the signal base.

**Parameters**

<i>nid</i>	The node to look at.
------------	----------------------

Referenced by `Parsers::LayoutFile::OrigX()`.

**9.179.5.61 OrigY()**

```
float Parsers::TrackGraph::OrigY (  
    int nid ) const
```

Return the  $y$  coordinate of the signal base.



### Parameters

<i>nid</i>	The node to look at.
------------	----------------------

Referenced by Parsers::LayoutFile::OrigY().

#### 9.179.5.62 ReverseActionScript()

```
const char* Parsers::TrackGraph::ReverseActionScript (
    int nid ) const
```

Return a block's or switchmotor's reverse action script.

Referenced by Parsers::LayoutFile::ReverseActionScript().

#### 9.179.5.63 Roots()

```
const IntegerList* Parsers::TrackGraph::Roots ( ) [inline]
```

Compressed graph roots.

Referenced by Parsers::LayoutFile::Roots().

#### 9.179.5.64 SenseScript()

```
const char* Parsers::TrackGraph::SenseScript (
    int nid ) const
```

Return a block's or switchmotor's sense script.

Referenced by Parsers::LayoutFile::SenseScript().

#### 9.179.5.65 SignalAspects()

```
const StringPairList* Parsers::TrackGraph::SignalAspects (
    int nid ) const
```

Return a Signal's aspect list.

Referenced by Parsers::LayoutFile::SignalAspects().

**9.179.5.66 tr\_rotate()**

```
Transform2D* Parsers::TrackGraph::tr_rotate (
    float amount,
    RotationUnit measure ) [private]
```

Construct a rotational transform.

**9.179.5.67 tr\_scale()** [1/2]

```
Transform2D* Parsers::TrackGraph::tr_scale (
    float mag_factor ) [private]
```

Construct a uniform scale transform.

**9.179.5.68 tr\_scale()** [2/2]

```
Transform2D* Parsers::TrackGraph::tr_scale (
    float xscale,
    float yscale ) [private]
```

Construct a non-uniform scale transform.

**9.179.5.69 tr\_translate()**

```
Transform2D* Parsers::TrackGraph::tr_translate (
    float x,
    float y ) [private]
```

Construct a translation transform.

**9.179.5.70 TrackList()**

```
const IntegerList* Parsers::TrackGraph::TrackList (
    int nid ) const
```

Return a block's tracklist.

Referenced by Parsers::LayoutFile::TrackList().

#### 9.179.5.71 traversePrimMST()

```
void Parsers::TrackGraph::traversePrimMST (
    CompressedEdgePairVector & result,
    CompressedNodeVector & parents,
    CompressedNode r ) const [private]
```

Traverse a PrimMST, starting at root *r*, inserting EdgePairs into result.

#### 9.179.5.72 TurnoutNumber()

```
int Parsers::TrackGraph::TurnoutNumber (
    int nid ) const
```

Return a switchmotor's turnout number.

Referenced by Parsers::LayoutFile::TurnoutNumber().

#### 9.179.5.73 TypeOfNode()

```
NodeType Parsers::TrackGraph::TypeOfNode (
    int nid ) const
```

Returns the type of the node.

Referenced by Parsers::LayoutFile::TypeOfNode().

### 9.179.6 Friends And Related Function Documentation

#### 9.179.6.1 operator<<

```
std::ostream& operator<< (
    ostream & stream,
    TrackGraph & graph ) [friend]
```

Output operator.

### 9.179.7 Member Data Documentation

#### 9.179.7.1 backpointers

```
std::map<Node, CompressedNode> Parsers::TrackGraph::backpointers [private]
```

Backpointer map.

#### 9.179.7.2 c\_idMap

```
CompressedIdNodeMap Parsers::TrackGraph::c_idMap [private]
```

Node Id map.

#### 9.179.7.3 c\_nodes

```
CompressedGraph Parsers::TrackGraph::c_nodes [private]
```

Compressed Graph adjacency\_list.

#### 9.179.7.4 c\_roots

```
IntegerList* Parsers::TrackGraph::c_roots [private]
```

Compressed Graph Roots.

#### 9.179.7.5 circleLayoutP

```
bool Parsers::TrackGraph::circleLayoutP [private]
```

Has CompressedGraphCircleLayout been run?

#### 9.179.7.6 compressedP

```
bool Parsers::TrackGraph::compressedP [private]
```

Is graph compressed?

#### 9.179.7.7 heads

`IntegerList*` Parsers::TrackGraph::heads [private]

Uncompressed graph heads (strong components).

#### 9.179.7.8 idMap

`IdNodeMap` Parsers::TrackGraph::idMap [private]

Node Id map.

#### 9.179.7.9 KamadaKawaiSpringLayoutP

`bool` Parsers::TrackGraph::KamadaKawaiSpringLayoutP [private]

Has CompressedGraphKamadaKawaiSpring been run?

#### 9.179.7.10 nodes

`Graph` Parsers::TrackGraph::nodes [private]

Graph adjacency\_list.

#### 9.179.7.11 none

`Node` Parsers::TrackGraph::none [private]

Special node that is nowhere (where all unconnected trackage goes).

#### 9.179.7.12 valid\_heads

`bool` Parsers::TrackGraph::valid\_heads [private]

Flag to indicate if heads is valid.

## 9.180 TTSupport::Train Class Reference

This class implements a train.

```
#include <Train.h>
```

### Public Member Functions

- [Train](#) ([TimeTableSystem](#) \*timetable=NULL, string [name](#)="", string [number](#)="", int [speed](#)=0, int [classnumber](#)=0, int [departure](#)=0, int start=0, int end=-1)  
*Create and initialize a train object.*
- const char \* [Name](#) () const  
*Return the name of the train.*
- const char \* [Number](#) () const  
*Return the number (or symbol) of the train.*
- int [Departure](#) () const  
*Return the departure time.*
- void [SetDeparture](#) (int depart)  
*Update departure time.*
- int [Speed](#) () const  
*Return the train's speed.*
- int [ClassNumber](#) () const  
*Return the class number.*
- int [NumberOfNotes](#) () const  
*Number of notes.*
- int [Note](#) (int i) const  
*Return the ith note.*
- void [AddNoteToTrain](#) (int note)  
*Add a note.*
- void [RemoveNoteFromTrain](#) (int note)  
*Remove a note.*
- void [UpdateStopLayover](#) (int istop, double layover)  
*Update stop layover.*
- void [UpdateStopCab](#) (int istop, [Cab](#) \*cab)  
*Update the cab.*
- void [AddNoteToStop](#) (int istop, int note)  
*Add a note to a stop.*
- void [RemoveNoteFromStop](#) (int istop, int note)  
*Remove a note from a stop.*
- void [SetOriginStorageTrack](#) (string trackname)  
*Set the origin storage track.*
- void [SetDestinationStorageTrack](#) (string trackname)  
*Set the destination storage track.*
- void [SetTransitStorageTrack](#) (int istop, string trackname)  
*Set an intermediate storage track.*
- int [NumberOfStops](#) () const

*Return the number of stops.*

- const [Stop](#) \* [StopI](#) (int i) const

*Return the ith stop object.*

- double [StartSMile](#) () const

*Return the start Scale Mile.*

- ostream & [Write](#) (ostream &stream) const

*Write object to a stream.*

- istream & [Read](#) (istream &stream, const [CabNameMap](#) cabs)

*Read an object from a stream.*

## Private Attributes

- string [name](#)

*Name of the train.*

- string [number](#)

*Number or symbol of the train;.*

- int [speed](#)

*The train's speed.*

- int [classnumber](#)

*The train's class.*

- vector< int > [notes](#)

*Notes about the train.*

- int [departure](#)

*Departure time.*

- [StopVector](#) [stops](#)

*The train's stops.*

- double [startSMile](#)

*Start scale mile.*

### 9.180.1 Detailed Description

This class implements a train.

A train travels down the track passing or stoping at stations along the way.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.180.2 Constructor & Destructor Documentation

### 9.180.2.1 Train()

```
TTSupport::Train::Train (
    TimeTableSystem * timetable = NULL,
    string name = "",
    string number = "",
    int speed = 0,
    int classnumber = 0,
    int departure = 0,
    int start = 0,
    int end = -1 )
```

Create and initialize a train object.

#### Parameters

<i>timetable</i>	The parent time table object.
<i>name</i>	The name of the train.
<i>number</i>	The number (or symbol) of the train.
<i>speed</i>	The maximum (scale) speed of the train.
<i>classnumber</i>	The class of the train.
<i>departure</i>	The train's departure time.
<i>start</i>	The originating station index.
<i>end</i>	The terminating station index.

## 9.180.3 Member Function Documentation

### 9.180.3.1 AddNoteToStop()

```
void TTSupport::Train::AddNoteToStop (
    int istop,
    int note )
```

Add a note to a stop.

#### Parameters

<i>istop</i>	The stop number to update.
<i>note</i>	The note to add.



### 9.180.3.2 AddNoteToTrain()

```
void TTSupport::Train::AddNoteToTrain (
    int note ) [inline]
```

Add a note.

#### Parameters

<i>note</i>	The note number to add.
-------------	-------------------------

References `i`, and `TTSupport::Stop::notes`.

### 9.180.3.3 ClassNumber()

```
int TTSupport::Train::ClassNumber ( ) const [inline]
```

Return the class number.

### 9.180.3.4 Departure()

```
int TTSupport::Train::Departure ( ) const [inline]
```

Return the departure time.

### 9.180.3.5 Name()

```
const char* TTSupport::Train::Name ( ) const [inline]
```

Return the name of the train.

### 9.180.3.6 Note()

```
int TTSupport::Train::Note (
    int i ) const [inline]
```

Return the *i*th note.

Returns -1 if the index is out of range.

**Parameters**

<i>i</i>	The index of the note.
----------	------------------------

References `i`, and `TTSupport::Stop::notes`.

**9.180.3.7 Number()**

```
const char* TTSupport::Train::Number ( ) const [inline]
```

Return the number (or symbol) of the train.

**9.180.3.8 NumberOfNotes()**

```
int TTSupport::Train::NumberOfNotes ( ) const [inline]
```

Number of notes.

References `TTSupport::Stop::notes`.

**9.180.3.9 NumberOfStops()**

```
int TTSupport::Train::NumberOfStops ( ) const [inline]
```

Return the number of stops.

**9.180.3.10 Read()**

```
istream& TTSupport::Train::Read (
    istream & stream,
    const CabNameMap cabs )
```

Read an object from a stream.

**Parameters**

<i>stream</i>	Stream to read from.
<i>cabs</i>	The cab name map.

#### 9.180.3.11 RemoveNoteFromStop()

```
void TTSupport::Train::RemoveNoteFromStop (
    int istop,
    int note )
```

Remove a note from a stop.

##### Parameters

<i>istop</i>	The stop number to update.
<i>note</i>	The note to remove.

#### 9.180.3.12 RemoveNoteFromTrain()

```
void TTSupport::Train::RemoveNoteFromTrain (
    int note ) [inline]
```

Remove a note.

##### Parameters

<i>note</i>	The note number to remove.
-------------	----------------------------

References TTSupport::Stop::cab, TTSupport::Stop::layover, and TTSupport::Stop::notes.

#### 9.180.3.13 SetDeparture()

```
void TTSupport::Train::SetDeparture (
    int depart ) [inline]
```

Update departure time.

##### Parameters

<i>depart</i>	The new departure time.
---------------	-------------------------

**9.180.3.14 SetDestinationStorageTrack()**

```
void TTSupport::Train::SetDestinationStorageTrack (
    string trackname )
```

Set the destination storage track.

**Parameters**

<i>trackname</i>	The terminating storage track name.
------------------	-------------------------------------

**9.180.3.15 SetOriginStorageTrack()**

```
void TTSupport::Train::SetOriginStorageTrack (
    string trackname )
```

Set the origin storage track.

**Parameters**

<i>trackname</i>	The originating storage track name.
------------------	-------------------------------------

**9.180.3.16 SetTransitStorageTrack()**

```
void TTSupport::Train::SetTransitStorageTrack (
    int istop,
    string trackname )
```

Set an intermediate storage track.

**Parameters**

<i>istop</i>	The stop index.
<i>trackname</i>	The intermediate storage track name.

**9.180.3.17 Speed()**

```
int TTSupport::Train::Speed ( ) const [inline]
```

Return the train's speed.

#### 9.180.3.18 StartSMile()

```
double TTSupport::Train::StartSMile ( ) const [inline]
```

Return the start Scale Mile.

References TTSupport::Stop::Read(), lcc::stream, and TTSupport::Stop::Write().

#### 9.180.3.19 StopI()

```
const Stop* TTSupport::Train::StopI (
    int i ) const [inline]
```

Return the ith stop object.

Returns NULL if the index is out of range.

##### Parameters

<i>i</i>	The index of the stop.
----------	------------------------

References i.

#### 9.180.3.20 UpdateStopCab()

```
void TTSupport::Train::UpdateStopCab (
    int istop,
    Cab * cab )
```

Update the cab.

##### Parameters

<i>istop</i>	The stop number to update.
<i>cab</i>	The new cab.

#### 9.180.3.21 UpdateStopLayover()

```
void TTSupport::Train::UpdateStopLayover (
    int istop,
    double layover )
```

Update stop layover.

##### Parameters

<i>istop</i>	The stop number to update.
<i>layover</i>	The new layover time.

#### 9.180.3.22 Write()

```
ostream& TTSupport::Train::Write (
    ostream & stream ) const
```

Write object to a stream.

##### Parameters

<i>stream</i>	Stream to write to.
---------------	---------------------

### 9.180.4 Member Data Documentation

#### 9.180.4.1 classnumber

```
int TTSupport::Train::classnumber [private]
```

The train's class.

#### 9.180.4.2 departure

```
int TTSupport::Train::departure [private]
```

Departure time.

#### 9.180.4.3 name

```
string TTSupport::Train::name [private]
```

Name of the train.

#### 9.180.4.4 notes

```
vector<int> TTSupport::Train::notes [private]
```

Notes about the train.

#### 9.180.4.5 number

```
string TTSupport::Train::number [private]
```

Number or symbol of the train;.

#### 9.180.4.6 speed

```
int TTSupport::Train::speed [private]
```

The train's speed.

#### 9.180.4.7 startSMile

```
double TTSupport::Train::startSMile [private]
```

Start scale mile.

#### 9.180.4.8 stops

```
StopVector TTSupport::Train::stops [private]
```

The train's stops.

## 9.181 FCFSupport::Train Class Reference

The [Train](#) class represents a train.

```
#include <Train.h>
```

### Classes

- union [StationOrIndustry](#)  
*Union of stations or industries, used for stops.*

### Public Types

- enum [TrainType](#) {  
    [Unknown](#) = 0, [Wayfreight](#) = 'W', [BoxMove](#) = 'B', [Manifest](#) = 'M',  
    [Passenger](#) = 'P' }  
*Types of trains.*

### Public Member Functions

- [Train](#) ()  
*Default constructor.*
- [Train](#) ([Train](#) &other)  
*Copy construtor.*
- [Train](#) & [operator=](#) ([Train](#) &other)  
*Assignment operator.*
- [Train](#) (const char \*n, const char \*dl, const char \*ct, const char \*descr, int sh, int mc, int mcl, int mw, int ml, int od, bool p, bool d, [TrainType](#) t)  
*Full constructor.*
- [~Train](#) ()  
*Descructor.*
- const char \* [Name](#) () const  
*Return the train's name.*
- const char \* [DivisionList](#) () const  
*Return the train's division list (string of symbols).*
- const char \* [CarTypes](#) () const  
*Return the train's car type list (string of char type characters).*
- const char \* [Description](#) () const  
*Return the train's description.*
- int [Shift](#) () const  
*Return the train's shift.*
- void [SetShift](#) (int newshift)  
*Set the train's shift.*
- int [MaxCars](#) () const  
*Return the train's maximum number of cars.*



- int [MaxClear](#) () const  
*Return the train's maximum clearance plate.*
- int [MaxWeight](#) () const  
*Return the train's maximum weight class.*
- void [SetMaxWeight](#) (int newmaxweight)  
*Set the train's maximum weight class.*
- int [MaxLength](#) () const  
*Return the train's maximum length.*
- void [SetMaxLength](#) (int newmaxlength)  
*Set the train's maximum length.*
- int [OnDuty](#) () const  
*Return the train's on duty time, in minutes since midnight.*
- bool [Print](#) () const  
*Return the train's print flag.*
- void [SetPrint](#) (bool flag)  
*Set the train's print flag.*
- bool [Done](#) () const  
*Return the train's done flag.*
- [TrainType](#) [Type](#) () const  
*Return the train's type.*
- int [NumberOfOrders](#) () const  
*Return the number of train orders for this train.*
- const char \* [Order](#) (int index) const  
*Return the lth order.*
- int [NumberOfStops](#) () const  
*Return the number of stops this train makes.*
- [FCFSupport::Industry](#) \* [IndustryStop](#) (int index) const  
*Return the lth industry stop this train makes.*
- [Station](#) \* [StationStop](#) (int index) const  
*Return the lth station stop this train makes.*

## Private Attributes

- vector< string > [orders](#)  
*List of train orders.*
- vector< [StationOrIndustry](#) > [stops](#)  
*List of stops.*
- string [name](#)  
*Name of the train.*
- string [divList](#)  
*The list of division symbols for this train.*
- string [carTypes](#)  
*The list of car type charactes.*
- string [description](#)  
*The description of the train.*
- int [shift](#)

- The train's shift.*
- int [maxcars](#)  
*The maximum number of cars on this train.*
- int [maxclear](#)  
*The maximum clearance plate for this train.*
- int [maxweight](#)  
*The maximum weight class for this train.*
- int [maxlength](#)  
*The maximum length for this train.*
- int [onduty](#)  
*The onduty time for this train, in minutes since midnight.*
- bool [print](#)  
*The print flag for this train.*
- bool [done](#)  
*The done flag for this train.*
- [TrainType](#) type  
*The type of this train.*

## Friends

- class [System](#)  
*The [System](#) class is a friend.*

### 9.181.1 Detailed Description

The [Train](#) class represents a train.

A train has a name, a type, a description, a list of divisions it operates in, it takes a specific set of car types, operates during a specific shift (or possibly all shifts if it is a box movement), a set of stops it makes, an on duty time, a maximum number of cars, a maximum clearance plate, a maximum weight class, a maximum length and several flags.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.181.2 Member Enumeration Documentation

#### 9.181.2.1 TrainType

enum [FCFSupport::Train::TrainType](#)

Types of trains.

## Enumerator

Unknown	An unknown type of train.
Wayfreight	A Way Freight train.
BoxMove	A Box Move train.
Manifest	A Manifest Freight train.
Passenger	A Passenger train.

## 9.181.3 Constructor &amp; Destructor Documentation

## 9.181.3.1 Train() [1/3]

```
FCFSupport::Train::Train ( ) [inline]
```

Default constructor.

Initialize all slots to empty values.

References carTypes, description, divList, done, maxcars, maxclear, maxlength, maxweight, name, onduty, print, shift, type, and Unknown.

## 9.181.3.2 Train() [2/3]

```
FCFSupport::Train::Train (
    Train & other ) [inline]
```

Copy construtor.

Copy initial values from another instance.

## Parameters

<i>other</i>	The other <a href="#">Train</a> instance.
--------------	---

References carTypes, description, divList, done, maxcars, maxclear, maxlength, maxweight, name, onduty, orders, print, shift, stops, and type.

**9.181.3.3 Train()** [3/3]

```

FCFSupport::Train::Train (
    const char * n,
    const char * dl,
    const char * ct,
    const char * descr,
    int sh,
    int mc,
    int mcl,
    int mw,
    int ml,
    int od,
    bool p,
    bool d,
    TrainType t ) [inline]

```

Full constructor.

Initialize the class instance from a set of parameters.

**Parameters**

<i>n</i>	The new train's name.
<i>dl</i>	The new train's division list.
<i>ct</i>	The new train's car type list.
<i>descr</i>	The New train's description.
<i>sh</i>	The new train's shift.
<i>mc</i>	The new train's maximum car limit.
<i>mcl</i>	The new train's maximum clearance plate.
<i>mw</i>	The new train's maximum weight class.
<i>ml</i>	The new train's maximum length.
<i>od</i>	The new train's on duty time (in minutes since midnight).
<i>p</i>	A flag to indicate if a pickup / dropoff sheet should be printed for this train.
<i>d</i>	A flag to indicate if this train is done.
<i>t</i>	The new train's type.

References carTypes, description, divList, done, maxcars, maxclear, maxlength, maxweight, name, onduty, print, shift, and type.

**9.181.3.4 ~Train()**

```

FCFSupport::Train::~~Train ( ) [inline]

```

Descructor.

### 9.181.4 Member Function Documentation

#### 9.181.4.1 CarTypes()

```
const char* FCFSupport::Train::CarTypes ( ) const [inline]
```

Return the train's car type list (string of char type characters).

References carTypes.

#### 9.181.4.2 Description()

```
const char* FCFSupport::Train::Description ( ) const [inline]
```

Return the train's description.

References description.

#### 9.181.4.3 DivisionList()

```
const char* FCFSupport::Train::DivisionList ( ) const [inline]
```

Return the train's division list (string of symbols).

References divList.

#### 9.181.4.4 Done()

```
bool FCFSupport::Train::Done ( ) const [inline]
```

Return the train's done flag.

References done.

#### 9.181.4.5 IndustryStop()

```
FCFSupport::Industry* FCFSupport::Train::IndustryStop (
    int index ) const [inline]
```

Return the lth industry stop this train makes.

**Parameters**

<i>index</i>	The index of the the stop to retrieve.
--------------	--

References Manifest, stops, and type.

Referenced by FCFSupport::SwitchListElement::DropStopEQ().

**9.181.4.6 MaxCars()**

```
int FCFSupport::Train::MaxCars ( ) const [inline]
```

Return the train's maximum number of cars.

References maxcars.

**9.181.4.7 MaxClear()**

```
int FCFSupport::Train::MaxClear ( ) const [inline]
```

Return the train's maximum clearance plate.

References maxclear.

**9.181.4.8 MaxLength()**

```
int FCFSupport::Train::MaxLength ( ) const [inline]
```

Return the train's maximum length.

References maxlength.

**9.181.4.9 MaxWeight()**

```
int FCFSupport::Train::MaxWeight ( ) const [inline]
```

Return the train's maximum weight class.

References maxweight.

#### 9.181.4.10 Name()

```
const char* FCFSupport::Train::Name ( ) const [inline]
```

Return the train's name.

References name.

#### 9.181.4.11 NumberOfOrders()

```
int FCFSupport::Train::NumberOfOrders ( ) const [inline]
```

Return the number of train orders for this train.

References orders.

#### 9.181.4.12 NumberOfStops()

```
int FCFSupport::Train::NumberOfStops ( ) const [inline]
```

Return the number of stops this train makes.

References stops.

#### 9.181.4.13 OnDuty()

```
int FCFSupport::Train::OnDuty ( ) const [inline]
```

Return the train's on duty time, in minutes since midnight.

References onduty.

#### 9.181.4.14 operator=()

```
Train& FCFSupport::Train::operator= (
    Train & other ) [inline]
```

Assignment operator.

Copy values from another instance.

**Parameters**

<i>other</i>	The other <a href="#">Train</a> instance.
--------------	---

References carTypes, description, divList, done, maxcars, maxclear, maxlength, maxweight, name, onduty, orders, print, shift, stops, and type.

**9.181.4.15 Order()**

```
const char* FCFSupport::Train::Order (
    int index ) const [inline]
```

Return the lth order.

**Parameters**

<i>index</i>	The index of the order to retrieve.
--------------	-------------------------------------

References orders.

**9.181.4.16 Print()**

```
bool FCFSupport::Train::Print ( ) const [inline]
```

Return the train's print flag.

References print.

**9.181.4.17 SetMaxLength()**

```
void FCFSupport::Train::SetMaxLength (
    int newmaxlength ) [inline]
```

Set the train's maximum length.

**Parameters**

<i>newmaxlength</i>	New maximum length.
---------------------	---------------------



References maxlength.

#### 9.181.4.18 SetMaxWeight()

```
void FCFSupport::Train::SetMaxWeight (
    int newmaxweight ) [inline]
```

Set the train's maximum weight class.

##### Parameters

<i>newmaxweight</i>	New maximum weight class.
---------------------	---------------------------

References maxweight.

#### 9.181.4.19 SetPrint()

```
void FCFSupport::Train::SetPrint (
    bool flag ) [inline]
```

Set the train's print flag.

##### Parameters

<i>flag</i>	The new flag value.
-------------	---------------------

References print.

#### 9.181.4.20 SetShift()

```
void FCFSupport::Train::SetShift (
    int newshift ) [inline]
```

Set the train's shift.

##### Parameters

<i>newshift</i>	The new shift.
-----------------	----------------

References shift.

#### 9.181.4.21 Shift()

```
int FCFSupport::Train::Shift ( ) const [inline]
```

Return the train's shift.

References shift.

#### 9.181.4.22 StationStop()

```
Station* FCFSupport::Train::StationStop (
    int index ) const [inline]
```

Return the lth station stop this train makes.

##### Parameters

<i>index</i>	The index of the the stop to retrieve.
--------------	--

References Manifest, stops, and type.

Referenced by FCFSupport::SwitchListElement::DropStopEQ().

#### 9.181.4.23 Type()

```
TrainType FCFSupport::Train::Type ( ) const [inline]
```

Return the train's type.

References type.

Referenced by FCFSupport::SwitchListElement::DropStopEQ(), FCFSupport::SwitchListElement::DropStopIndustry(), and FCFSupport::SwitchListElement::DropStopStation().

### 9.181.5 Friends And Related Function Documentation

### 9.181.5.1 System

```
friend class System [friend]
```

The [System](#) class is a friend.

## 9.181.6 Member Data Documentation

### 9.181.6.1 carTypes

```
string FCFSupport::Train::carTypes [private]
```

The list of car type charactes.

Referenced by [CarTypes\(\)](#), [operator=\(\)](#), and [Train\(\)](#).

### 9.181.6.2 description

```
string FCFSupport::Train::description [private]
```

The description of the train.

Referenced by [Description\(\)](#), [operator=\(\)](#), and [Train\(\)](#).

### 9.181.6.3 divList

```
string FCFSupport::Train::divList [private]
```

The list of division symbols for this train.

Referenced by [DivisionList\(\)](#), [operator=\(\)](#), and [Train\(\)](#).

### 9.181.6.4 done

```
bool FCFSupport::Train::done [private]
```

The done flag for this train.

Referenced by [Done\(\)](#), [operator=\(\)](#), and [Train\(\)](#).

**9.181.6.5 maxcars**

```
int FCFSupport::Train::maxcars [private]
```

The maximum number of cars on this train.

Referenced by MaxCars(), operator=(), and Train().

**9.181.6.6 maxclear**

```
int FCFSupport::Train::maxclear [private]
```

The maximum clearance plate for this train.

Referenced by MaxClear(), operator=(), and Train().

**9.181.6.7 maxlength**

```
int FCFSupport::Train::maxlength [private]
```

The maximum length for this train.

Referenced by MaxLength(), operator=(), SetMaxLength(), and Train().

**9.181.6.8 maxweight**

```
int FCFSupport::Train::maxweight [private]
```

The maximum weight class for this train.

Referenced by MaxWeight(), operator=(), SetMaxWeight(), and Train().

**9.181.6.9 name**

```
string FCFSupport::Train::name [private]
```

Name of the train.

Referenced by Name(), operator=(), and Train().

#### 9.181.6.10 onduty

```
int FCFSupport::Train::onduty [private]
```

The onduty time for this train, in minutes since midnight.

Referenced by OnDuty(), operator=(), and Train().

#### 9.181.6.11 orders

```
vector<string> FCFSupport::Train::orders [private]
```

List of train orders.

Referenced by NumberOfOrders(), operator=(), Order(), and Train().

#### 9.181.6.12 print

```
bool FCFSupport::Train::print [private]
```

The print flag for this train.

Referenced by operator=(), Print(), SetPrint(), and Train().

#### 9.181.6.13 shift

```
int FCFSupport::Train::shift [private]
```

The train's shift.

Referenced by operator=(), SetShift(), Shift(), and Train().

#### 9.181.6.14 stops

```
vector<StationOrIndustry> FCFSupport::Train::stops [private]
```

List of stops.

Referenced by IndustryStop(), NumberOfStops(), operator=(), StationStop(), and Train().

### 9.181.6.15 type

```
TrainType FCFSupport::Train::type [private]
```

The type of this train.

Referenced by IndustryStop(), operator=(), StationStop(), Train(), and Type().

## 9.182 FCFSupport::TrainDisplayCallback Class Reference

Callback to manage a train status display.

```
#include <CallBack.h>
```

### Public Member Functions

- [TrainDisplayCallback](#) ()  
*Constructor.*
- virtual [~TrainDisplayCallback](#) ()  
*Destructor.*
- virtual void [InitializeTrainDisplay](#) (string name, int stationCount, int maxLength, int maxCars) const  
*Initialize the train status display.*
- virtual void [CloseTrainDisplay](#) () const  
*Close the train display.*
- virtual void [GrabTrainDisplay](#) () const  
*Grab the train display.*
- virtual void [ReleaseTrainDisplay](#) () const  
*Release the train display.*
- virtual void [UpdateTrainDisplay](#) (string currentStationName, string currentStopName, int trainLength, int number↔ Cars, int trainTons, int trainLoads, int trainEmpties, int trainLongest, int currentStop) const  
*Update the train display.*

### 9.182.1 Detailed Description

Callback to manage a train status display.

This callback is used to manage an application supplied train status display. Used in the train running methods when train runs are simulated to move cars from place place. The train status shows the train's progress and the pickups and drops it makes as it traverses its route.

```
@author Robert Heller \<heller\@deepsoft.com\>
```

### 9.182.2 Constructor & Destructor Documentation

### 9.182.2.1 TrainDisplayCallback()

```
FCFSupport::TrainDisplayCallback::TrainDisplayCallback ( ) [inline]
```

Constructor.

The base constructor does nothing. It is presumed that a derived class might do something useful.

### 9.182.2.2 ~TrainDisplayCallback()

```
virtual FCFSupport::TrainDisplayCallback::~~TrainDisplayCallback ( ) [inline], [virtual]
```

Destructor.

The base destructor does nothing. It is presumed that a derived class might do something useful.

## 9.182.3 Member Function Documentation

### 9.182.3.1 CloseTrainDisplay()

```
virtual void FCFSupport::TrainDisplayCallback::CloseTrainDisplay ( ) const [inline], [virtual]
```

Close the train display.

This is called when the train status display is no longer needed.

### 9.182.3.2 GrabTrainDisplay()

```
virtual void FCFSupport::TrainDisplayCallback::GrabTrainDisplay ( ) const [inline], [virtual]
```

Grab the train display.

This is used when the train status display needs to be “front and center”.

### 9.182.3.3 InitializeTrainDisplay()

```
virtual void FCFSupport::TrainDisplayCallback::InitializeTrainDisplay (
    string name,
    int stationCount,
    int maxLength,
    int maxCars ) const [inline], [virtual]
```

Initialize the train status display.

Set the train name, the station count, max length and the maximum number of cars. Generally, this initializes the train status display for a new train start.

**Parameters**

<i>name</i>	Name of the train.
<i>stationCount</i>	The station count (number of stops).
<i>maxLength</i>	Maximum train length.
<i>maxCars</i>	Maximum number of cars.

**9.182.3.4 ReleaseTrainDisplay()**

```
virtual void FCFSupport::TrainDisplayCallback::ReleaseTrainDisplay ( ) const [inline], [virtual]
```

Release the train display.

This is used when the train status display no longer needs to be "front and center".

**9.182.3.5 UpdateTrainDisplay()**

```
virtual void FCFSupport::TrainDisplayCallback::UpdateTrainDisplay (
    string currentStationName,
    string currentStopName,
    int trainLength,
    int numberCars,
    int trainTons,
    int trainLoads,
    int trainEmpties,
    int trainLongest,
    int currentStop ) const [inline], [virtual]
```

Update the train display.

This updates the train status display when a train arrives at a station (or industry), drops cars, picks up cars and leaves a station (or industry).

**Parameters**

<i>currentStationName</i>	The current station name.
<i>currentStopName</i>	The current stop name.
<i>trainLength</i>	The current train length.
<i>numberCars</i>	The current number of cars.
<i>trainTons</i>	The current number of tons.
<i>trainLoads</i>	The current number of loaded cars.
<i>trainEmpties</i>	The current number of empty cars.
<i>trainLongest</i>	The longest the train has been.
<i>currentStop</i>	The current stop number.



## 9.183 Parsers::TrackGraph::Transform2D Class Reference

Two dimensional transform class.

```
#include <TrackGraph.h>
```

### Public Member Functions

- [Transform2D](#) ()  
*Default constructor.*
- [Transform2D](#) (float r11, float r12, float tx, float r21, float r22, float ty, float a0=0.0, float a1=0.0, float s=1.0)  
*Full fledged constructor.*
- [Transform2D](#) (const [Transform2D](#) \*ts)  
*Copy constructor.*
- float [Determinant](#) () const  
*Return the determinant.*
- float [Minor](#) (int, int) const  
*Return the minor.*
- [Transform2D](#) \* [Inverse](#) () const  
*Return the inverse.*
- void [Apply](#) (float x, float y, float s, float &tx, float &ty, float &ts) const  
*Apply a scaled transformation.*
- int [Apply](#) (float x, float y, float &tx, float &ty) const  
*Apply a normal transformation/.*
- int [operator==](#) (const [Transform2D](#) &other) const  
*Equality operator.*
- int [operator!=](#) (const [Transform2D](#) &other) const  
*Inequality operator.*

### Private Attributes

- float [matrix](#) [3][3]  
*Transform matrix.*

### Static Private Attributes

- static constexpr float [FUZZ](#) = .00001  
*Fuzz factor.*

### Friends

- [Transform2D](#) \* [operator\\*](#) (const [Transform2D](#) &t1, const [Transform2D](#) &t2)  
*Matrix multiplication.*

### 9.183.1 Detailed Description

Two dimensional transform class.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.183.2 Constructor & Destructor Documentation

#### 9.183.2.1 Transform2D() [1/3]

```
Parsers::TrackGraph::Transform2D::Transform2D ( )
```

Default constructor.

Creates an identity tranform.

#### 9.183.2.2 Transform2D() [2/3]

```
Parsers::TrackGraph::Transform2D::Transform2D (
    float r11,
    float r12,
    float tx,
    float r21,
    float r22,
    float ty,
    float a0 = 0.0,
    float a1 = 0.0,
    float s = 1.0 )
```

Full fledged constructor.

#### 9.183.2.3 Transform2D() [3/3]

```
Parsers::TrackGraph::Transform2D::Transform2D (
    const Transform2D * ts )
```

Copy constructor.

### 9.183.3 Member Function Documentation

#### 9.183.3.1 Apply() [1/2]

```
void Parsers::TrackGraph::Transform2D::Apply (
    float x,
    float y,
    float s,
    float & tx,
    float & ty,
    float & ts ) const
```

Apply a scaled transformation.

#### 9.183.3.2 Apply() [2/2]

```
int Parsers::TrackGraph::Transform2D::Apply (
    float x,
    float y,
    float & tx,
    float & ty ) const
```

Apply a normal transformation/.

#### 9.183.3.3 Determinant()

```
float Parsers::TrackGraph::Transform2D::Determinant ( ) const
```

Return the determinant.

#### 9.183.3.4 Inverse()

```
Transform2D* Parsers::TrackGraph::Transform2D::Inverse ( ) const
```

Return the inverse.

#### 9.183.3.5 Minor()

```
float Parsers::TrackGraph::Transform2D::Minor (
    int ,
    int ) const
```

Return the minor.

#### 9.183.3.6 operator!=(())

```
int Parsers::TrackGraph::Transform2D::operator!= (
    const Transform2D & other ) const [inline]
```

Inequality operator.

#### 9.183.3.7 operator==(())

```
int Parsers::TrackGraph::Transform2D::operator== (
    const Transform2D & other ) const
```

Equality operator.

### 9.183.4 Friends And Related Function Documentation

#### 9.183.4.1 operator\*

```
Transform2D* operator* (
    const Transform2D & t1,
    const Transform2D & t2 ) [friend]
```

Matrix multiplication.

### 9.183.5 Member Data Documentation

## 9.183.5.1 FUZZ

```
constexpr float Parsers::TrackGraph::Transform2D::FUZZ = .00001 [static], [private]
```

Fuzz factor.

## 9.183.5.2 matrix

```
float Parsers::TrackGraph::Transform2D::matrix[3][3] [private]
```

Transform matrix.

## 9.184 Parsers::TurnoutBody Class Reference

List of turnout body lines (T, E, P, S, C, and J lines).

```
#include <TurnoutBody.h>
```

### Public Member Functions

- [TurnoutBody](#) ([TurnoutBodyElt](#) \*e, [TurnoutBody](#) \*n)  
*Basic constructor.*
- [TrackBody](#) \* [TurnoutEnds](#) ()  
*Create a track endpoint list.*
- int [TurnoutSegmentCount](#) ()  
*Count segments (S, C, and J lines).*
- int [TurnoutRouteCount](#) ()  
*Count routes (P lines).*
- const [TurnoutBodyElt](#) \* [Element](#) () const  
*Return current element.*

### Static Public Member Functions

- static [TurnoutBody](#) \* [ConsTurnoutBody](#) ([TurnoutBodyElt](#) \*trbe, [TurnoutBody](#) \*trb)  
*Alternative constructor function.*
- static void [CleanUpTurnoutBody](#) ([TurnoutBody](#) \*trb)  
*Free up memory.*

### Private Member Functions

- void [CleanUpElement](#) ()  
*Free up memory.*

## Private Attributes

- [TurnoutBodyElt](#) \* [element](#)  
*Current element.*
- [TurnoutBody](#) \* [next](#)  
*Next element.*

## Friends

- class [TurnoutBodyElt](#)
- class [TrackGraph](#)

### 9.184.1 Detailed Description

List of turnout body lines (T, E, P, S, C, and J lines).

#### Author

Robert Heller <heller@deepsoft.com>

### 9.184.2 Constructor & Destructor Documentation

#### 9.184.2.1 TurnoutBody()

```
Parsers::TurnoutBody::TurnoutBody (
    TurnoutBodyElt * e,
    TurnoutBody * n ) [inline]
```

Basic constructor.

### 9.184.3 Member Function Documentation

#### 9.184.3.1 CleanupElement()

```
void Parsers::TurnoutBody::CleanupElement ( ) [inline], [private]
```

Free up memory.

References [Parsers::TurnoutBodyElt::theEnd](#), and [Parsers::TurnoutBodyElt::theType](#).

Referenced by [CleanupTurnoutBody\(\)](#).

### 9.184.3.2 CleanupTurnoutBody()

```
static void Parsers::TurnoutBody::CleanupTurnoutBody (
    TurnoutBody * trb ) [inline], [static]
```

Free up memory.

References CleanupElement(), element, and next.

### 9.184.3.3 ConsTurnoutBody()

```
static TurnoutBody* Parsers::TurnoutBody::ConsTurnoutBody (
    TurnoutBodyElt * trbe,
    TurnoutBody * trb ) [inline], [static]
```

Alternative constructor function.

### 9.184.3.4 Element()

```
const TurnoutBodyElt* Parsers::TurnoutBody::Element ( ) const [inline]
```

Return current element.

### 9.184.3.5 TurnoutEnds()

```
TrackBody* Parsers::TurnoutBody::TurnoutEnds ( ) [inline]
```

Create a track endpoint list.

References element, next, Parsers::TurnoutBodyElt::theEnd, and Parsers::TurnoutBodyElt::theType.

### 9.184.3.6 TurnoutRouteCount()

```
int Parsers::TurnoutBody::TurnoutRouteCount ( ) [inline]
```

Count routes (P lines).

References element, next, and Parsers::TurnoutBodyElt::theType.

#### 9.184.3.7 TurnoutSegmentCount()

```
int Parsers::TurnoutBody::TurnoutSegmentCount ( ) [inline]
```

Count segments (S, C, and J lines).

References element, next, and Parsers::TurnoutBodyElt::theType.

### 9.184.4 Friends And Related Function Documentation

#### 9.184.4.1 TrackGraph

```
friend class TrackGraph [friend]
```

#### 9.184.4.2 TurnoutBodyElt

```
friend class TurnoutBodyElt [friend]
```

### 9.184.5 Member Data Documentation

#### 9.184.5.1 element

```
TurnoutBodyElt* Parsers::TurnoutBody::element [private]
```

Current element.

Referenced by CleanUpTurnoutBody(), TurnoutEnds(), TurnoutRouteCount(), and TurnoutSegmentCount().

#### 9.184.5.2 next

```
TurnoutBody* Parsers::TurnoutBody::next [private]
```

Next element.

Referenced by CleanUpTurnoutBody(), TurnoutEnds(), TurnoutRouteCount(), and TurnoutSegmentCount().



## 9.185 Parsers::TurnoutBodyElt Class Reference

Turnout body elements: T, E, P, S, C, and J lines are collected.

```
#include <TurnoutBody.h>
```

### Classes

- struct [Pos](#)  
*Position structure.*

### Public Types

- enum [TurnoutBodyEltType](#) {  
[None](#), [TurnoutEnd](#), [TurnoutRoute](#), [TurnoutStraightSegment](#),  
[TurnoutCurveSegment](#), [TurnoutJointSegment](#) }  
*Element types.*

### Public Member Functions

- [TurnoutBodyElt](#) ()  
*Constructor.*
- [~TurnoutBodyElt](#) ()  
*Destructor.*
- [TurnoutBodyEltType TheType](#) () const  
*Type accessor.*
- void [GetTurnoutRoute](#) (char \*&pName, [IntegerList](#) \*&cList) const  
*Fetch turnout route data.*
- int [GetTurnoutStraightSegment](#) (float &x1, float &y1, float &x2, float &y2) const  
*Fetch turnout straight segment data.*
- int [GetTurnoutCurveSegment](#) (float &r, float &x, float &y, float &a0, float &a1) const  
*Fetch turnout curve segment data.*
- int [GetTurnoutJointSegment](#) (float &x, float &y, float &a, float &l0, float &l1, float &r, float &l) const  
*Fetch turnout joint segment data.*

### Static Public Member Functions

- static void [InitTSegId](#) ()  
*Segment count initializer.*
- static [TurnoutBodyElt](#) \* [MakeTurnoutEnd](#) ([TrackBodyElt](#) \*tbe)  
*Create an endpoint (T or E lines).*
- static [TurnoutBodyElt](#) \* [MakeTurnoutRoute](#) (char \*pName, [IntegerList](#) \*cList)  
*Create a turnout route (P lines).*
- static [TurnoutBodyElt](#) \* [MakeTurnoutStraightSegment](#) (float x1, float y1, float x2, float y2)  
*Create a turnout straight segment (S lines).*
- static [TurnoutBodyElt](#) \* [MakeTurnoutCurveSegment](#) (float r, float x, float y, float a0, float a1)  
*Create a turnout curve segment (C lines).*
- static [TurnoutBodyElt](#) \* [MakeTurnoutJointSegment](#) (float x, float y, float a, float l0, float l1, float r, float l)  
*Create a turnout joint segment (J lines).*

## Private Attributes

- [TurnoutBodyEltType](#) *theType*  
*Element type.*
- [TrackBodyElt](#) \* [theEnd](#)  
*Pointer to T or E line data.*
- char \* [RouteName](#)  
*Route name (P lines).*
- [IntegerList](#) \* [routeList](#)  
*Segment list (P Lines).*
- int [segmentId](#)  
*Segment index (S, C, or J lines).*
- [Pos](#) [pos1](#)  
*First position.*
- [Pos](#) [pos2](#)  
*Second position.*
- float [radius](#)  
*A radius value.*
- float [ang0](#)  
*An angle value.*
- float [ang1](#)  
*Another angle value.*
- float [R](#)  
*\$R\$ value (for J lines).*
- float [L](#)  
*\$L\$ value (for J lines).*

## Static Private Attributes

- static int [segCount](#)  
*Counter for S, C, and J segments.*

## Friends

- class [TrackGraph](#)
- class [TurnoutBody](#)

### 9.185.1 Detailed Description

Turnout body elements: T, E, P, S, C, and J lines are collected.

others are discarded.

#### Author

Robert Heller <heller@deepsoft.com>

## 9.185.2 Member Enumeration Documentation

### 9.185.2.1 TurnoutBodyEltType

enum `Parsers::TurnoutBodyElt::TurnoutBodyEltType`

Element types.

Enumerator

None	Placeholder.
TurnoutEnd	T or E line.
TurnoutRoute	P line.
TurnoutStraightSegment	S line.
TurnoutCurveSegment	C line.
TurnoutJointSegment	J Line.

## 9.185.3 Constructor & Destructor Documentation

### 9.185.3.1 TurnoutBodyElt()

`Parsers::TurnoutBodyElt::TurnoutBodyElt ( ) [inline]`

Constructor.

### 9.185.3.2 ~TurnoutBodyElt()

`Parsers::TurnoutBodyElt::~~TurnoutBodyElt ( ) [inline]`

Destructor.

## 9.185.4 Member Function Documentation

#### 9.185.4.1 GetTurnoutCurveSegment()

```
int Parsers::TurnoutBodyElt::GetTurnoutCurveSegment (
    float & r,
    float & x,
    float & y,
    float & a0,
    float & a1 ) const [inline]
```

Fetch turnout curve segment data.

References Parsers::TurnoutBodyElt::Pos::x, and Parsers::TurnoutBodyElt::Pos::y.

#### 9.185.4.2 GetTurnoutJointSegment()

```
int Parsers::TurnoutBodyElt::GetTurnoutJointSegment (
    float & x,
    float & y,
    float & a,
    float & l0,
    float & l1,
    float & r,
    float & l ) const [inline]
```

Fetch turnout joint segment data.

References angle, len0, len1, Parsers::TurnoutBodyElt::Pos::x, and Parsers::TurnoutBodyElt::Pos::y.

#### 9.185.4.3 GetTurnoutRoute()

```
void Parsers::TurnoutBodyElt::GetTurnoutRoute (
    char *& pName,
    IntegerList *& cList ) const [inline]
```

Fetch turnout route data.

#### 9.185.4.4 GetTurnoutStraightSegment()

```
int Parsers::TurnoutBodyElt::GetTurnoutStraightSegment (
    float & x1,
    float & y1,
    float & x2,
    float & y2 ) const [inline]
```

Fetch turnout straight segment data.

References Parsers::TurnoutBodyElt::Pos::x, and Parsers::TurnoutBodyElt::Pos::y.

#### 9.185.4.5 InitTSegId()

```
static void Parsers::TurnoutBodyElt::InitTSegId ( ) [inline], [static]
```

Segment count initializer.

#### 9.185.4.6 MakeTurnoutCurveSegment()

```
static TurnoutBodyElt* Parsers::TurnoutBodyElt::MakeTurnoutCurveSegment (
    float r,
    float x,
    float y,
    float a0,
    float a1 ) [inline], [static]
```

Create a turnout curve segment (C lines).

References ang0, ang1, pos1, radius, segmentId, theType, Parsers::TurnoutBodyElt::Pos::x, and Parsers::TurnoutBodyElt::Pos::y.

#### 9.185.4.7 MakeTurnoutEnd()

```
static TurnoutBodyElt* Parsers::TurnoutBodyElt::MakeTurnoutEnd (
    TrackBodyElt * tbe ) [inline], [static]
```

Create an endpoint (T or E lines).

References theEnd, and theType.

#### 9.185.4.8 MakeTurnoutJointSegment()

```
static TurnoutBodyElt* Parsers::TurnoutBodyElt::MakeTurnoutJointSegment (
    float x,
    float y,
    float a,
    float l0,
    float l1,
    float r,
    float l ) [inline], [static]
```

Create a turnout joint segment (J lines).

References L, pos1, R, segmentId, theType, Parsers::TurnoutBodyElt::Pos::x, and Parsers::TurnoutBodyElt::Pos::y.

#### 9.185.4.9 MakeTurnoutRoute()

```
static TurnoutBodyElt* Parsers::TurnoutBodyElt::MakeTurnoutRoute (
    char * pName,
    IntegerList * cList ) [inline], [static]
```

Create a turnout route (P lines).

References routeList, RouteName, and theType.

#### 9.185.4.10 MakeTurnoutStraightSegment()

```
static TurnoutBodyElt* Parsers::TurnoutBodyElt::MakeTurnoutStraightSegment (
    float x1,
    float y1,
    float x2,
    float y2 ) [inline], [static]
```

Create a turnout straight segment (S lines).

References pos1, pos2, segmentId, theType, Parsers::TurnoutBodyElt::Pos::x, and Parsers::TurnoutBodyElt::Pos::y.

#### 9.185.4.11 TheType()

```
TurnoutBodyEltType Parsers::TurnoutBodyElt::TheType ( ) const [inline]
```

Type accessor.

### 9.185.5 Friends And Related Function Documentation

#### 9.185.5.1 TrackGraph

```
friend class TrackGraph [friend]
```

#### 9.185.5.2 TurnoutBody

```
friend class TurnoutBody [friend]
```

## 9.185.6 Member Data Documentation

### 9.185.6.1 ang0

```
float Parsers::TurnoutBodyElt::ang0 [private]
```

An angle value.

Referenced by `MakeTurnoutCurveSegment()`.

### 9.185.6.2 ang1

```
float Parsers::TurnoutBodyElt::ang1 [private]
```

Another angle value.

Referenced by `MakeTurnoutCurveSegment()`.

### 9.185.6.3 L

```
float Parsers::TurnoutBodyElt::L [private]
```

\$L\$ value (for J lines).

Referenced by `MakeTurnoutJointSegment()`.

### 9.185.6.4 pos1

```
Pos Parsers::TurnoutBodyElt::pos1 [private]
```

First position.

Referenced by `MakeTurnoutCurveSegment()`, `MakeTurnoutJointSegment()`, and `MakeTurnoutStraightSegment()`.

**9.185.6.5 pos2**

```
Pos Parsers::TurnoutBodyElt::pos2 [private]
```

Second position.

Referenced by MakeTurnoutStraightSegment().

**9.185.6.6 R**

```
float Parsers::TurnoutBodyElt::R [private]
```

\$R\$ value (for J lines).

Referenced by MakeTurnoutJointSegment().

**9.185.6.7 radius**

```
float Parsers::TurnoutBodyElt::radius [private]
```

A radius value.

Referenced by MakeTurnoutCurveSegment().

**9.185.6.8 routeList**

```
IntegerList* Parsers::TurnoutBodyElt::routeList [private]
```

Segment list (P Lines).

Referenced by MakeTurnoutRoute().

**9.185.6.9 RouteName**

```
char* Parsers::TurnoutBodyElt::RouteName [private]
```

Route name (P lines).

Referenced by MakeTurnoutRoute().



#### 9.185.6.10 segCount

```
int Parsers::TurnoutBodyElt::segCount [static], [private]
```

Counter for S, C, and J segments.

#### 9.185.6.11 segmentId

```
int Parsers::TurnoutBodyElt::segmentId [private]
```

Segment index (S, C, or J lines).

Referenced by MakeTurnoutCurveSegment(), MakeTurnoutJointSegment(), and MakeTurnoutStraightSegment().

#### 9.185.6.12 theEnd

```
TrackBodyElt* Parsers::TurnoutBodyElt::theEnd [private]
```

Pointer to T or E line data.

Referenced by Parsers::TurnoutBody::CleanUpElement(), MakeTurnoutEnd(), and Parsers::TurnoutBody::TurnoutEnds().

#### 9.185.6.13 theType

```
TurnoutBodyEltType Parsers::TurnoutBodyElt::theType [private]
```

Element type.

Referenced by Parsers::TurnoutBody::CleanUpElement(), MakeTurnoutCurveSegment(), MakeTurnoutEnd(), MakeTurnoutJointSegment(), MakeTurnoutRoute(), MakeTurnoutStraightSegment(), Parsers::TurnoutBody::TurnoutEnds(), Parsers::TurnoutBody::TurnoutRouteCount(), and Parsers::TurnoutBody::TurnoutSegmentCount().

## 9.186 Parsers::TurnoutGraphic Struct Reference

Structure holding a turnout's graphical information.

```
#include <TrackGraph.h>
```

## Public Attributes

- float [minX](#)  
*Minimum \$X\$ coordinate.*
- float [minY](#)  
*Minimum \$Y\$ coordinate.*
- float [maxX](#)  
*Maximum \$X\$ coordinate.*
- float [maxY](#)  
*Maximum \$Y\$ coordinate.*
- int [numSegments](#)  
*Number of segments.*
- [SegVector](#) \* [segments](#)  
*Segment vector.*

### 9.186.1 Detailed Description

Structure holding a turnout's graphical information.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.186.2 Member Data Documentation

#### 9.186.2.1 [maxX](#)

```
float Parsers::TurnoutGraphic::maxX
```

Maximum \$X\$ coordinate.

#### 9.186.2.2 [maxY](#)

```
float Parsers::TurnoutGraphic::maxY
```

Maximum \$Y\$ coordinate.

#### 9.186.2.3 minX

```
float Parsers::TurnoutGraphic::minX
```

Minimum \$X\$ coordinate.

#### 9.186.2.4 minY

```
float Parsers::TurnoutGraphic::minY
```

Minimum \$Y\$ coordinate.

#### 9.186.2.5 numSegments

```
int Parsers::TurnoutGraphic::numSegments
```

Number of segments.

#### 9.186.2.6 segments

```
SegVector* Parsers::TurnoutGraphic::segments
```

Segment vector.

### 9.187 Parsers::TurnoutRoutelist Struct Reference

Turnout route list structure.

```
#include <TrackGraph.h>
```

#### Public Attributes

- int [numRoutelists](#)  
*Number of routes.*
- [RouteVec](#) \* [routes](#)  
*Route vector.*

### 9.187.1 Detailed Description

Turnout route list structure.

Author

Robert Heller <heller@deepsoft.com>

### 9.187.2 Member Data Documentation

#### 9.187.2.1 numRoutelists

```
int Parsers::TurnoutRoutelist::numRoutelists
```

Number of routes.

#### 9.187.2.2 routes

```
RouteVec* Parsers::TurnoutRoutelist::routes
```

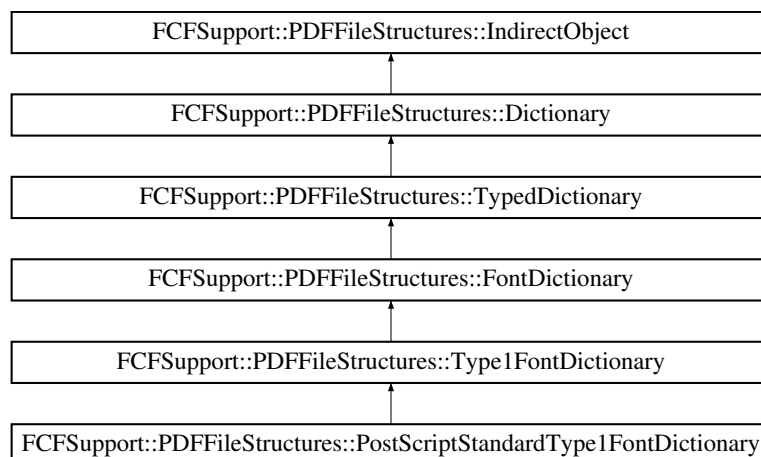
Route vector.

## 9.188 FCFSupport::PDFFFileStructures::Type1FontDictionary Class Reference

Type 1 Font dictionary.

```
#include <PDFPrinterSupport.h>
```

Inheritance diagram for FCFSupport::PDFFFileStructures::Type1FontDictionary:



## Public Member Functions

- [Type1FontDictionary](#) (const string basefont, int firstchar, int lastchar, [IndirectFloatVector](#) \*widths\_, [TypedDictionary](#) \*fontdescriptor, const string encoding="", unsigned long int objNum=0L, unsigned short int genNum=0, [CrossReferenceTable](#) \*tab=NULL)  
*Constructor.*
- [Type1FontDictionary](#) (const string basefont, int firstchar, int lastchar, [IndirectFloatVector](#) \*widths\_, [TypedDictionary](#) \*fontdescriptor, [TypedDictionary](#) \*encoding, unsigned long int objNum=0L, unsigned short int genNum=0, [CrossReferenceTable](#) \*tab=NULL)  
*Constructor.*
- [~Type1FontDictionary](#) ()  
*Destructor.*

## Protected Member Functions

- virtual ostream & [WriteDictionaryElements](#) (ostream &stream) const  
*Write an object directly.*

## Private Attributes

- string [baseFont](#)  
*Base font name.*
- int [firstChar](#)  
*First character in widths array;.*
- int [lastChar](#)  
*Last character in widths array.*
- [IndirectFloatVector](#) \* [widths](#)  
*Widths array.*
- [TypedDictionary](#) \* [fontDescriptor](#)  
*Font Descriptor.*
- string [encodingName](#)  
*Encoding as a name.*
- [TypedDictionary](#) \* [encodingDictionary](#)  
*Encoding as a dictionary.*

### 9.188.1 Detailed Description

Type 1 Font dictionary.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.188.2 Constructor & Destructor Documentation

### 9.188.2.1 Type1FontDictionary() [1/2]

```
FCFSupport::PDFFileStructures::Type1FontDictionary::Type1FontDictionary (
    const string basefont,
    int firstchar,
    int lastchar,
    IndirectFloatVector * widths_,
    TypedDictionary * fontdescriptor,
    const string encoding = "",
    unsigned long int objNum = 0L,
    unsigned short int genNum = 0,
    CrossReferenceTable * tab = NULL ) [inline]
```

Constructor.

Build a Type 1 font.

#### Parameters

<i>basefont</i>	Name of the base font.
<i>firstchar</i>	The first character code.
<i>lastchar</i>	The last character code.
<i>widths_</i>	The widths of the characters.
<i>fontdescriptor</i>	The font description.
<i>encoding</i>	The encoding of the font.
<i>objNum</i>	The next object number.
<i>genNum</i>	The generation number.
<i>tab</i>	The cross reference table we are in.

### 9.188.2.2 Type1FontDictionary() [2/2]

```
FCFSupport::PDFFileStructures::Type1FontDictionary::Type1FontDictionary (
    const string basefont,
    int firstchar,
    int lastchar,
    IndirectFloatVector * widths_,
    TypedDictionary * fontdescriptor,
    TypedDictionary * encoding,
    unsigned long int objNum = 0L,
    unsigned short int genNum = 0,
    CrossReferenceTable * tab = NULL ) [inline]
```

Constructor.

Build a Type 1 font.

## Parameters

<i>basefont</i>	Name of the base font.
<i>firstchar</i>	The first character code.
<i>lastchar</i>	The last character code.
<i>widths_</i>	The widths of the characters.
<i>fontdescriptor</i>	The font description.
<i>encoding</i>	The encoding of the font.
<i>objNum</i>	The next object number.
<i>genNum</i>	The generation number.
<i>tab</i>	The cross reference table we are in.

## 9.188.2.3 ~Type1FontDictionary()

```
FCFSupport::PDFFileStructures::Type1FontDictionary::~Type1FontDictionary ( ) [inline]
```

Destructor.

References lcc::stream.

## 9.188.3 Member Function Documentation

## 9.188.3.1 WriteDictionaryElements()

```
virtual ostream& FCFSupport::PDFFileStructures::Type1FontDictionary::WriteDictionaryElements (
    ostream & stream ) const [protected], [virtual]
```

Write an object directly.

## Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

Reimplemented from [FCFSupport::PDFFileStructures::FontDictionary](#).

## 9.188.4 Member Data Documentation

**9.188.4.1 baseFont**

```
string FCFSupport::PDFFileStructures::Type1FontDictionary::baseFont [private]
```

Base font name.

**9.188.4.2 encodingDictionary**

```
TypedDictionary* FCFSupport::PDFFileStructures::Type1FontDictionary::encodingDictionary [private]
```

Encoding as a dictionary.

**9.188.4.3 encodingName**

```
string FCFSupport::PDFFileStructures::Type1FontDictionary::encodingName [private]
```

Encoding as a name.

**9.188.4.4 firstChar**

```
int FCFSupport::PDFFileStructures::Type1FontDictionary::firstChar [private]
```

First character in widths array;.

**9.188.4.5 fontDescriptor**

```
TypedDictionary* FCFSupport::PDFFileStructures::Type1FontDictionary::fontDescriptor [private]
```

Font Descriptor.

**9.188.4.6 lastChar**

```
int FCFSupport::PDFFileStructures::Type1FontDictionary::lastChar [private]
```

Last character in widths array.



## 9.188.4.7 widths

```
IndirectFloatVector* FCFSupport::PDFFileStructures::Type1FontDictionary::widths [private]
```

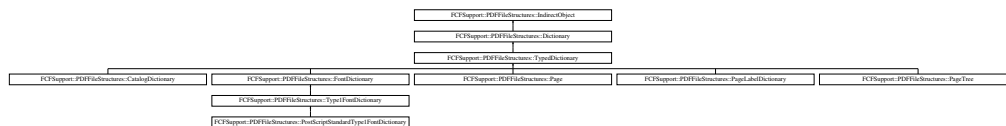
Widths array.

## 9.189 FCFSupport::PDFFileStructures::TypedDictionary Class Reference

Typed dictionary.

```
#include <PDFPrinterSupport.h>
```

Inheritance diagram for FCFSupport::PDFFileStructures::TypedDictionary:



### Public Member Functions

- [TypedDictionary](#) (string t, unsigned long int objNum=0L, unsigned short int genNum=0, CrossReferenceTable \*tab=NULL)  
*Constructor.*
- [~TypedDictionary](#) ()  
*Destructor.*

### Protected Member Functions

- ostream & [WriteDictionaryType](#) (ostream &stream) const  
*Write the dictionary's type.*
- virtual ostream & [WriteDictionaryElements](#) (ostream &stream) const  
*Write this dictionary's elements.*

### Private Attributes

- string [type](#)  
*The dictionary's type name.*

### 9.189.1 Detailed Description

Typed dictionary.

A dictionary with a /Type field.

Author

Robert Heller <heller@deepsoft.com>

### 9.189.2 Constructor & Destructor Documentation

#### 9.189.2.1 TypedDictionary()

```
FCFSupport::PDFFileStructures::TypedDictionary::TypedDictionary (
    string t,
    unsigned long int objNum = 0L,
    unsigned short int genNum = 0,
    CrossReferenceTable * tab = NULL ) [inline]
```

Constructor.

Set the type member.

Parameters

<i>t</i>	The type of this dictionary.
<i>objNum</i>	The next object number.
<i>genNum</i>	The generation number.
<i>tab</i>	The cross reference table we are in.

#### 9.189.2.2 ~TypedDictionary()

```
FCFSupport::PDFFileStructures::TypedDictionary::~~TypedDictionary ( ) [inline]
```

Destructor.

### 9.189.3 Member Function Documentation

### 9.189.3.1 WriteDictionaryElements()

```
virtual ostream& FCFSupport::PDFFileStructures::TypedDictionary::WriteDictionaryElements (
    ostream & stream ) const [inline], [protected], [virtual]
```

Write this dictionary's elements.

Start with its type.

#### Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

Reimplemented from [FCFSupport::PDFFileStructures::Dictionary](#).

Reimplemented in [FCFSupport::PDFFileStructures::CatalogDictionary](#), [FCFSupport::PDFFileStructures::Type1FontDictionary](#), [FCFSupport::PDFFileStructures::FontDictionary](#), [FCFSupport::PDFFileStructures::PageLabelDictionary](#), [FCFSupport::PDFFileStructures::PageTree](#), and [FCFSupport::PDFFileStructures::Page](#).

### 9.189.3.2 WriteDictionaryType()

```
ostream& FCFSupport::PDFFileStructures::TypedDictionary::WriteDictionaryType (
    ostream & stream ) const [inline], [protected]
```

Write the dictionary's type.

#### Parameters

<i>stream</i>	The output stream to write to.
---------------	--------------------------------

## 9.189.4 Member Data Documentation

### 9.189.4.1 type

```
string FCFSupport::PDFFileStructures::TypedDictionary::type [private]
```

The dictionary's type name.

## 9.190 FCFSupport::WorkInProgressCallback Class Reference

Work In Progress Callback.

```
#include <CallBack.h>
```

### Public Member Functions

- [WorkInProgressCallback](#) ()  
*Constructor.*
- virtual [~WorkInProgressCallback](#) ()  
*Destructor.*
- virtual void [ProgressStart](#) (const string Message) const  
*Start up the work in progress display.*
- virtual void [ProgressUpdate](#) (int Percent, const string Message) const  
*Update the progress meter.*
- virtual void [ProgressDone](#) (const string Message) const  
*Mark the process meter as done.*

### 9.190.1 Detailed Description

Work In Progress Callback.

Provides a callback to manage a work in progress display. This class is a dummy base class. Applications can define member functions that manage an application specific work in progress display.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.190.2 Constructor & Destructor Documentation

#### 9.190.2.1 WorkInProgressCallback()

```
FCFSupport::WorkInProgressCallback::WorkInProgressCallback ( ) [inline]
```

Constructor.

The base constructor does nothing. It is presumed that a derived class might do something useful.

### 9.190.2.2 ~WorkInProgressCallback()

```
virtual FCFSupport::WorkInProgressCallback::~WorkInProgressCallback ( ) [inline], [virtual]
```

Destructor.

The base destructor does nothing. It is presumed that a derived class might do something useful.

## 9.190.3 Member Function Documentation

### 9.190.3.1 ProgressDone()

```
virtual void FCFSupport::WorkInProgressCallback::ProgressDone (
    const string Message ) const [inline], [virtual]
```

Mark the process meter as done.

Forces the meter to 100 percent and display a work completion message.

Parameters

<i>Message</i>	A message to display.
----------------	-----------------------

Referenced by FCFSupport::PauseCallback::Pause().

### 9.190.3.2 ProgressStart()

```
virtual void FCFSupport::WorkInProgressCallback::ProgressStart (
    const string Message ) const [inline], [virtual]
```

Start up the work in progress display.

An initial message is passed to be displayed.

Parameters

<i>Message</i>	An initial message string.
----------------	----------------------------

Referenced by FCFSupport::PauseCallback::Pause().

### 9.190.3.3 ProgressUpdate()

```
virtual void FCFSupport::WorkInProgressCallback::ProgressUpdate (
    int Percent,
    const string Message ) const [inline], [virtual]
```

Update the progress meter.

Advance the progress meter to the percent completed and display an updated message describing the progress.

#### Parameters

<i>Percent</i>	The completion percentage, between 0 and 100. A value of 100 indicates that the job is done.
<i>Message</i>	A message to display, typically something identifying what tasks have been completed.

Referenced by FCFSupport::PauseCallback::Pause().

## 9.191 xpressnet::XPressNet Class Reference

Main [XPressNet](#) interface class.

### Public Member Functions

- [XPressNet](#) (name, port="/dev/ttyS0")  
*The constructor opens the serial port and initializes the port.*
- [~XPressNet](#) ()  
*The destructor restores the serial port's state and closes it.*
- [CheckForResponse](#) (timeout=5)  
*Check for a response from the command station.*
- [GetNextCommandStationResponse](#) (timeout=5)  
*Return the next response from the command station.*
- [ResumeOperations](#) ()  
*Resume operations request.*
- [StopOperations](#) ()  
*Stop operations request.*
- [EmergencyStopAllLocomotives](#) ()  
*Emergency stop all locomotives.*
- [EmergencyStopALocomotive](#) (la)  
*Emergency stop a locomotive.*
- [RegisterModeRead](#) (r)  
*Register mode read.*
- [DirectModeCVRead](#) (cv)  
*Direct mode CV read.*
- [PagedModeCVRead](#) (cv)

- Paged mode CV read.*
- [RequestForServiceModeResults](#) ()  
*Request for service mode results.*
- [RegisterModeWrite](#) (r, d)  
*Register mode write.*
- [DirectModeCVWrite](#) (cv, d)  
*Direct mode CV write.*
- [PagedModeCVWrite](#) (cv, d)  
*Paged mode CV write.*
- [CommandStationSoftwareVersion](#) ()  
*Command station software version request.*
- [CommandStationStatusRequest](#) ()  
*Command station status request.*
- [SetCommandStationPowerUpMode](#) (mode)  
*Set command station power up mode.*
- [AccessoryDecoderInformationRequest](#) (address, nibble)  
*Accessory decoder information request.*
- [AccessoryDecoderOperation](#) (groupaddr, elementaddr, activateOutput, useOutput2)  
*Accessory decoder operation request.*
- [LocomotiveInformationRequest](#) (address)  
*Locomotive information request.*
- [FunctionStatusRequest](#) (address)  
*Function status request.*
- [SetLocomotiveSpeedAndDirection](#) (address, ssm, dir, speed)  
*Set locomotive speed and direction.*
- [SetLocomotiveFunctionsGroup1](#) (address, f0, f1, f2, f3, f4)  
*Set locomotive functions, group 1.*
- [SetLocomotiveFunctionsGroup2](#) (address, f5, f6, f7, f8)  
*Set locomotive functions, group 2.*
- [SetLocomotiveFunctionsGroup3](#) (address, f9, f10, f11, f12)  
*Set locomotive functions, group 3.*
- [SetFunctionStateGroup1](#) (address, f0, f1, f2, f3, f4)  
*Set locomotive function state, group 1.*
- [SetFunctionStateGroup2](#) (address, f5, f6, f7, f8)  
*Set locomotive function state, group 2.*
- [SetFunctionStateGroup3](#) (address, f9, f10, f11, f12)  
*Set locomotive function state, group 3.*
- [EstablishDoubleHeader](#) (address1, address2)  
*Establish a double header.*
- [DissolveDoubleHeader](#) (address1)  
*Dissolve a double header.*
- [OperatingModeProgrammingByteModeWrite](#) (address, cv, data)  
*Operating mode programming byte mode write.*
- [OperatingModeProgrammingBitModeWrite](#) (address, cv, bitnum, value)  
*Operating mode programming bit mode write.*
- [AddLocomotiveToMultiUnit](#) (address, mtr, samedirection)  
*Add locomotive to Multi-Unit.*

- [RemoveLocomotiveFromMultiUnit](#) (address, mtr)  
*Remove locomotive to Multi-Unit.*
- [AddressInquiryNextMUMember](#) (mtr, address)  
*Address inquire next MU member.*
- [AddressInquiryPreviousMUMember](#) (mtr, address)  
*Address inquire previous MU member.*
- [AddressInquiryNextMU](#) (mtr)  
*Address inquire next MU.*
- [AddressInquiryPreviousMU](#) (mtr)  
*Address inquire previous MU.*
- [AddressInquiryNextStack](#) (address)  
*Address inquire next stack.*
- [AddressInquiryPreviousStack](#) (address)  
*Address inquire previous stack.*
- [DeleteLocomotiveFromStack](#) (address)  
*Delete locomotive from stack.*
- [GetLI100VersionNumbers](#) ()  
*Fetch version numbers from LI100F and LI101.*
- [SetLI101Address](#) (addr)  
*Set LI101's XPressNet address.*
- [readevent](#) (script)  
*Establish an external read event handler.*

### Private Member Functions

- [\\_appendXORByte](#) (messageVar)  
*Compute and append the XOR check byte.*
- [\\_readevent](#) ()  
*Read event handler, toggle timeout flag.*
- [\\_timeoutevent](#) ()  
*Timeout event handler, toggle timeout flag.*
- [\\_transmit](#) (themessage)  
*Transmit a message.*
- [\\_readbyte](#) (thebytevar, timeout)  
*Read next available byte or return false.*

### Static Private Member Functions

- static [\\_CheckForResponse\\_0x00](#) (message)  
*Helper function for CheckForResponse: handles the 0x00 arm.*
- static [\\_CheckForResponse\\_0x40](#) (message)  
*Helper function for CheckForResponse: handles the 0x40 arm.*
- static [\\_CheckForResponse\\_0x60](#) (message)  
*Helper function for CheckForResponse: handles the 0x60 arm.*
- static [\\_CheckForResponse\\_0x80](#) (message)  
*Helper function for CheckForResponse: handles the 0x80 arm.*



- static [\\_CheckForResponse\\_0xa0](#) (message)  
*Helper function for CheckForResponse: handles the 0xa0 arm.*
- static [\\_CheckForResponse\\_0xc0](#) (message)  
*Helper function for CheckForResponse: handles the 0xc0 arm.*
- static [\\_CheckForResponse\\_0xe0](#) (message)  
*Helper function for CheckForResponse: handles the 0xe0 arm.*
- static [\\_ADDRESS](#) (a)  
*Helper function to insure a proper address.*

## Private Attributes

- [ttyfd](#)  
*Terminal file descriptor.*
- [responseList](#)  
*Response list.*
- [\\_timeout](#)  
*Timeout or data available flag.*

### 9.191.1 Detailed Description

Main [XPressNet](#) interface class.

This class implements the interface logic to connect to the XpressNet.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.191.2 Constructor & Destructor Documentation

#### 9.191.2.1 XPressNet()

```
xpressnet::XPressNet::XPressNet (
    name ,
    port = "/dev/ttyS0" )
```

The constructor opens the serial port and initializes the port.

#### Parameters

<i>port</i>	The serial port device file.
-------------	------------------------------

### 9.191.2.2 ~XPressNet()

```
xpressnet::XPressNet::~~XPressNet ( )
```

The destructor restores the serial port's state and closes it.

## 9.191.3 Member Function Documentation

### 9.191.3.1 \_ADDRESS()

```
static xpressnet::XPressNet::_ADDRESS (
    a ) [static], [private]
```

Helper function to insure a proper address.

If it is a long address ( $\geq 100$ ), 0x0c000 is added.

#### Parameters

<i>a</i>	Raw address.
----------	--------------

### 9.191.3.2 \_appendXORByte()

```
xpressnet::XPressNet::_appendXORByte (
    messageVar ) [private]
```

Compute and append the XOR check byte.

#### Parameters

<i>messageVar</i>	Name of the list holding the message bytes.
-------------------	---

### 9.191.3.3 \_CheckForResponse\_0x00()

```
static xpressnet::XPressNet::_CheckForResponse_0x00 (
```

```
message ) [static], [private]
```

Helper function for CheckForResponse: handles the 0x00 arm.

#### 9.191.3.4 \_CheckForResponse\_0x40()

```
static xpressnet::XPressNet::_CheckForResponse_0x40 (  
    message ) [static], [private]
```

Helper function for CheckForResponse: handles the 0x40 arm.

#### 9.191.3.5 \_CheckForResponse\_0x60()

```
static xpressnet::XPressNet::_CheckForResponse_0x60 (  
    message ) [static], [private]
```

Helper function for CheckForResponse: handles the 0x60 arm.

#### 9.191.3.6 \_CheckForResponse\_0x80()

```
static xpressnet::XPressNet::_CheckForResponse_0x80 (  
    message ) [static], [private]
```

Helper function for CheckForResponse: handles the 0x80 arm.

#### 9.191.3.7 \_CheckForResponse\_0xa0()

```
static xpressnet::XPressNet::_CheckForResponse_0xa0 (  
    message ) [static], [private]
```

Helper function for CheckForResponse: handles the 0xa0 arm.

#### 9.191.3.8 \_CheckForResponse\_0xc0()

```
static xpressnet::XPressNet::_CheckForResponse_0xc0 (
    message ) [static], [private]
```

Helper function for CheckForResponse: handles the 0xc0 arm.

#### 9.191.3.9 \_CheckForResponse\_0xe0()

```
static xpressnet::XPressNet::_CheckForResponse_0xe0 (
    message ) [static], [private]
```

Helper function for CheckForResponse: handles the 0xe0 arm.

#### 9.191.3.10 \_readbyte()

```
xpressnet::XPressNet::_readbyte (
    thebytevar ,
    timeout ) [private]
```

Read next available byte or return false.

##### Parameters

<i>thebytevar</i>	Name of a variable to receive the byte.
<i>timeout</i>	Timeout in seconds.

If there is a defined external read event handler, the timeout parameter is ignored and false is returned if there are no bytes available. The presumption is that the read is being called from event handler and that means that there is data available.

#### 9.191.3.11 \_readevent()

```
xpressnet::XPressNet::_readevent ( ) [private]
```

Read event handler, toggle timeout flag.

#### 9.191.3.12 \_timeoutevent()

```
xpressnet::XPressNet::_timeoutevent ( ) [private]
```

Timeout event handler, toggle timeout flag.

#### 9.191.3.13 \_transmit()

```
xpressnet::XPressNet::_transmit (
    themessage ) [private]
```

Transmit a message.

#### 9.191.3.14 AccessoryDecoderInformationRequest()

```
xpressnet::XPressNet::AccessoryDecoderInformationRequest (
    address ,
    nibble )
```

Accessory decoder information request.

##### Parameters

<i>address</i>	Address of decoder.
<i>nibble</i>	Which nibble.

#### 9.191.3.15 AccessoryDecoderOperation()

```
xpressnet::XPressNet::AccessoryDecoderOperation (
    groupaddr ,
    elementaddr ,
    activateOutput ,
    useOutput2 )
```

Accessory decoder operation request.

##### Parameters

<i>groupaddr</i>	Address of decoder.
<i>elementaddr</i>	Address of element.
<i>activateOutput</i>	Set or clear output.
<i>useOutput2</i>	Use output 2?

### 9.191.3.16 AddLocomotiveToMultiUnit()

```
xpressnet::XPressNet::AddLocomotiveToMultiUnit (
    address ,
    mtr ,
    samedirection )
```

Add locomotive to Multi-Unit.

#### Parameters

<i>address</i>	Locomotive address.
<i>mtr</i>	Multi-Unit address.
<i>samedirection</i>	The locomotive direction is the same as the consist direction.

### 9.191.3.17 AddressInquiryNextMU()

```
xpressnet::XPressNet::AddressInquiryNextMU (
    mtr )
```

Address inquire next MU.

#### Parameters

<i>mtr</i>	Multi-Unit address.
------------	---------------------

### 9.191.3.18 AddressInquiryNextMUMember()

```
xpressnet::XPressNet::AddressInquiryNextMUMember (
    mtr ,
    address )
```

Address inquire next MU member.

#### Parameters

<i>mtr</i>	Multi-Unit address.
<i>address</i>	Locomotive address.

### 9.191.3.19 AddressInquiryNextStack()

```
xpressnet::XPressNet::AddressInquiryNextStack (
    address )
```

Address inquire next stack.

#### Parameters

<i>address</i>	Locomotive address.
----------------	---------------------

### 9.191.3.20 AddressInquiryPreviousMU()

```
xpressnet::XPressNet::AddressInquiryPreviousMU (
    mtr )
```

Address inquire previous MU.

#### Parameters

<i>mtr</i>	Multi-Unit address.
------------	---------------------

### 9.191.3.21 AddressInquiryPreviousMUMember()

```
xpressnet::XPressNet::AddressInquiryPreviousMUMember (
    mtr ,
    address )
```

Address inquire previous MU member.

#### Parameters

<i>mtr</i>	Multi-Unit address.
<i>address</i>	Locomotive address.

**9.191.3.22 AddressInquiryPreviousStack()**

```
xpressnet::XPressNet::AddressInquiryPreviousStack (
    address )
```

Address inquire previous stack.

**Parameters**

<i>address</i>	Locomotive address.
----------------	---------------------

**9.191.3.23 CheckForResponse()**

```
xpressnet::XPressNet::CheckForResponse (
    timeout = 5 )
```

Check for a response from the command station.

**Parameters**

<i>timeout</i>	Timeout in seconds
----------------	--------------------

**9.191.3.24 CommandStationSoftwareVersion()**

```
xpressnet::XPressNet::CommandStationSoftwareVersion ( )
```

Command station software version request.

**9.191.3.25 CommandStationStatusRequest()**

```
xpressnet::XPressNet::CommandStationStatusRequest ( )
```

Command station status request.

**9.191.3.26 DeleteLocomotiveFromStack()**

```
xpressnet::XPressNet::DeleteLocomotiveFromStack (
    address )
```

Delete locomotive from stack.



## Parameters

<i>address</i>	Locomotive address.
----------------	---------------------

## 9.191.3.27 DirectModeCVRead()

```
xpressnet::XPressNet::DirectModeCVRead (
    cv )
```

Direct mode CV read.

## Parameters

<i>cv</i>	CV to read.
-----------	-------------

## 9.191.3.28 DirectModeCVWrite()

```
xpressnet::XPressNet::DirectModeCVWrite (
    cv ,
    d )
```

Direct mode CV write.

## Parameters

<i>cv</i>	CV to write to.
<i>d</i>	Data to write.

## 9.191.3.29 DissolveDoubleHeader()

```
xpressnet::XPressNet::DissolveDoubleHeader (
    address1 )
```

Dissolve a double header.

## Parameters

<i>address1</i>	Locomotive address1.
-----------------	----------------------

**9.191.3.30 EmergencyStopAllLocomotives()**

```
xpressnet::XPressNet::EmergencyStopAllLocomotives ( )
```

Emergency stop all locomotives.

**9.191.3.31 EmergencyStopALocomotive()**

```
xpressnet::XPressNet::EmergencyStopALocomotive (
    la )
```

Emergency stop a locomotive.

**Parameters**

<i>la</i>	Address of the locomotive to stop.
-----------	------------------------------------

**9.191.3.32 EstablishDoubleHeader()**

```
xpressnet::XPressNet::EstablishDoubleHeader (
    address1 ,
    address2 )
```

Establish a double header.

**Parameters**

<i>address1</i>	Locomotive address1.
<i>address2</i>	Locomotive address2.

**9.191.3.33 FunctionStatusRequest()**

```
xpressnet::XPressNet::FunctionStatusRequest (
    address )
```

Function status request.

## Parameters

<i>address</i>	Address of locomotive.
----------------	------------------------

## 9.191.3.34 GetLI100VersionNumbers()

```
xpressnet::XPressNet::GetLI100VersionNumbers ( )
```

Fetch version numbers from LI100F and LI101.

## 9.191.3.35 GetNextCommandStationResponse()

```
xpressnet::XPressNet::GetNextCommandStationResponse (
    timeout = 5 )
```

Return the next response from the command station.

## Parameters

<i>timeout</i>	Timeout in seconds
----------------	--------------------

## 9.191.3.36 LocomotiveInformationRequest()

```
xpressnet::XPressNet::LocomotiveInformationRequest (
    address )
```

Locomotive information request.

## Parameters

<i>address</i>	Address of locomotive.
----------------	------------------------

## 9.191.3.37 OperatingModeProgrammingBitModeWrite()

```
xpressnet::XPressNet::OperatingModeProgrammingBitModeWrite (
    address ,
```

```
cv ,  
bitnum ,  
value )
```

Operating mode programming bit mode write.

#### Parameters

<i>address</i>	Locomotive address.
<i>cv</i>	CV to set.
<i>bitnum</i>	Bit number.
<i>value</i>	Value to set.

#### 9.191.3.38 OperatingModeProgrammingByteModeWrite()

```
xpressnet::XPressNet::OperatingModeProgrammingByteModeWrite (  
    address ,  
    cv ,  
    data )
```

Operating mode programming byte mode write.

#### Parameters

<i>address</i>	Locomotive address.
<i>cv</i>	CV to set.
<i>data</i>	Data to set.

#### 9.191.3.39 PagedModeCVRead()

```
xpressnet::XPressNet::PagedModeCVRead (  
    cv )
```

Paged mode CV read.

#### Parameters

<i>cv</i>	CV to read.
-----------	-------------

#### 9.191.3.40 PagedModeCVWrite()

```
xpressnet::XPressNet::PagedModeCVWrite (
    cv ,
    d )
```

Paged mode CV write.

##### Parameters

<i>cv</i>	CV to write to.
<i>d</i>	Data to write.

#### 9.191.3.41 readevent()

```
xpressnet::XPressNet::readevent (
    script )
```

Establish an external read event handler.

##### Parameters

<i>script</i>	The external event handler script.
---------------	------------------------------------

#### 9.191.3.42 RegisterModeRead()

```
xpressnet::XPressNet::RegisterModeRead (
    r )
```

Register mode read.

##### Parameters

<i>r</i>	Register to read.
----------	-------------------

#### 9.191.3.43 RegisterModeWrite()

```
xpressnet::XPressNet::RegisterModeWrite (
```

```
    r ,  
    d )
```

Register mode write.

#### Parameters

<i>r</i>	Register to write to.
<i>d</i>	Data to write.

#### 9.191.3.44 RemoveLocomotiveFromMultiUnit()

```
xpressnet::XPressNet::RemoveLocomotiveFromMultiUnit (   
    address ,  
    mtr )
```

Remove locomotive to Multi-Unit.

#### Parameters

<i>address</i>	Locomotive address.
<i>mtr</i>	Multi-Unit address.

#### 9.191.3.45 RequestForServiceModeResults()

```
xpressnet::XPressNet::RequestForServiceModeResults ( )
```

Request for service mode results.

#### 9.191.3.46 ResumeOperations()

```
xpressnet::XPressNet::ResumeOperations ( )
```

Resume operations request.

#### 9.191.3.47 SetCommandStationPowerUpMode()

```
xpressnet::XPressNet::SetCommandStationPowerUpMode (   
    mode )
```

Set command station power up mode.

## Parameters

<i>mode</i>	Mode to set.
-------------	--------------

## 9.191.3.48 SetFunctionStateGroup1()

```
xpressnet::XPressNet::SetFunctionStateGroup1 (
    address ,
    f0 ,
    f1 ,
    f2 ,
    f3 ,
    f4 )
```

Set locomotive function state, group 1.

## Parameters

<i>address</i>	Locomotive address.
<i>f0</i>	Function 0.
<i>f1</i>	Function 1.
<i>f2</i>	Function 2.
<i>f3</i>	Function 3.
<i>f4</i>	Function 4.

## 9.191.3.49 SetFunctionStateGroup2()

```
xpressnet::XPressNet::SetFunctionStateGroup2 (
    address ,
    f5 ,
    f6 ,
    f7 ,
    f8 )
```

Set locomotive function state, group 2.

## Parameters

<i>address</i>	Locomotive address.
<i>f5</i>	Function 5.
<i>f6</i>	Function 6.
<i>f7</i>	Function 7.
<i>f8</i>	Function 8.

**9.191.3.50 SetFunctionStateGroup3()**

```
xpressnet::XPressNet::SetFunctionStateGroup3 (
    address ,
    f9 ,
    f10 ,
    f11 ,
    f12 )
```

Set locomotive function state, group 3.

**Parameters**

<i>address</i>	Locomotive address.
<i>f9</i>	Function 9.
<i>f10</i>	Function 10.
<i>f11</i>	Function 11.
<i>f12</i>	Function 12.

**9.191.3.51 SetLI101Address()**

```
xpressnet::XPressNet::SetLI101Address (
    addr )
```

Set LI101's [XPressNet](#) address.

**9.191.3.52 SetLocomotiveFunctionsGroup1()**

```
xpressnet::XPressNet::SetLocomotiveFunctionsGroup1 (
    address ,
    f0 ,
    f1 ,
    f2 ,
    f3 ,
    f4 )
```

Set locomotive functions, group 1.

**Parameters**

<i>address</i>	Locomotive address.
<i>f0</i>	Function 0.
<i>f1</i>	Function 1.
<i>f2</i>	Function 2.
<i>f3</i>	Function 3.
<i>f4</i>	Function 4.



### 9.191.3.53 SetLocomotiveFunctionsGroup2()

```
xpressnet::XPressNet::SetLocomotiveFunctionsGroup2 (
    address ,
    f5 ,
    f6 ,
    f7 ,
    f8 )
```

Set locomotive functions, group 2.

#### Parameters

<i>address</i>	Locomotive address.
<i>f5</i>	Function 5.
<i>f6</i>	Function 6.
<i>f7</i>	Function 7.
<i>f8</i>	Function 8.

### 9.191.3.54 SetLocomotiveFunctionsGroup3()

```
xpressnet::XPressNet::SetLocomotiveFunctionsGroup3 (
    address ,
    f9 ,
    f10 ,
    f11 ,
    f12 )
```

Set locomotive functions, group 3.

#### Parameters

<i>address</i>	Locomotive address.
<i>f9</i>	Function 9.
<i>f10</i>	Function 10.
<i>f11</i>	Function 11.
<i>f12</i>	Function 12.

### 9.191.3.55 SetLocomotiveSpeedAndDirection()

```
xpressnet::XPressNet::SetLocomotiveSpeedAndDirection (
```

```
    address ,  
    ssm ,  
    dir ,  
    speed )
```

Set locomotive speed and direction.

#### Parameters

<i>address</i>	Address of locomotive.
<i>ssm</i>	Speed step mode to use.
<i>dir</i>	Desired direction.
<i>speed</i>	Desired speed.

#### 9.191.3.56 StopOperations()

```
xpressnet::XPressNet::StopOperations ( )
```

Stop operations request.

### 9.191.4 Member Data Documentation

#### 9.191.4.1 \_timeout

```
xpressnet::XPressNet::_timeout [private]
```

Timeout or data available flag.

#### 9.191.4.2 responseList

```
xpressnet::XPressNet::responseList [private]
```

Response list.

## 9.191.4.3 ttyfd

```
xpressnet::XpressNet::ttyfd [private]
```

Terminal file descriptor.

## 9.192 xpressnet::XpressNetEvent Class Reference

[XPressNet](#) Event class.

### Public Member Functions

- [XpressNetEvent](#) (name, script, port="/dev/ttyS0")  
*Constructor.*
- [~XpressNetEvent](#) ()  
*Destructor.*

### Private Member Functions

- [\\_eventhandler](#) ()  
*The event handler.*

### Private Attributes

- [xpressnet](#)  
*Holds the [XPressNet](#) component.*
- [\\_script](#)  
*Holds the event script.*

### 9.192.1 Detailed Description

[XPressNet](#) Event class.

This class implements the Tcl Event interface to the [XPressNet](#) serial port interface. A Tcl script is bound to [XPressNet](#) serial port events. This script is called from the event procedures when [XPressNet](#) events occur.

#### Author

Robert Heller <heller@deepsoft.com>

### 9.192.2 Constructor & Destructor Documentation

#### 9.192.2.1 XpressNetEvent()

```
xpressnet::XpressNetEvent::XpressNetEvent (
    name ,
    script ,
    port = "/dev/ttyS0" )
```

Constructor.

The constructor opens serial port and initializes the port, stashes the interpreter and creates an event source.

##### Parameters

<i>script</i>	The event script.
<i>port</i>	The serial port device file.

#### 9.192.2.2 ~XpressNetEvent()

```
xpressnet::XpressNetEvent::~XpressNetEvent ( )
```

Destructor.

The destructor closes the serial port and deletes the event source.

### 9.192.3 Member Function Documentation

#### 9.192.3.1 \_eventhandler()

```
xpressnet::XpressNetEvent::_eventhandler ( ) [private]
```

The event handler.

### 9.192.4 Member Data Documentation

## 9.192.4.1 \_script

```
xpressnet::XpressNetEvent::_script [private]
```

Holds the event script.

## 9.192.4.2 xpressnet

```
xpressnet::XpressNetEvent::xpressnet [private]
```

Holds the [XPressNet](#) component.

## 9.193 YY\_MRRXtrkCad\_INHERIT Class Reference

```
#include <MRRXtrkCad.tab.h>
```

## Public Types

- enum [YY\\_MRRXtrkCad\\_ENUM\\_TOKEN](#) {  
[YY\\_MRRXtrkCad\\_NULL\\_TOKEN](#) =0, [INTEGER](#) =258, [FLOAT](#) =259, [STRING](#) =260,  
[RESTOFLINE](#) =261, [MULTILINE](#) =262, [EOL](#) =263, [INTERMSTRING](#) =264,  
[NOTWORD](#) =265, [ENDSEGS](#) =266, [ENDSIGNAL](#) =267, [ENDBLOCK](#) =268,  
[ENDTRACKS](#) =269, [\\_VERSION](#) =270, [TITLE](#) =271, [MAPSCALE](#) =272,  
[ROOMSIZE](#) =273, [SCALE](#) =274, [HO](#) =275, [N](#) =276,  
[O](#) =277, [LAYERS](#) =278, [CURRENT](#) =279, [STRUCTURE](#) =280,  
[DRAW](#) =281, [BEZIER](#) =282, [BZRLIN](#) =283, [CORNU](#) =284,  
[SUBSEGS](#) =285, [SUBSEND](#) =286, [CURVE](#) =287, [TURNOUT](#) =288,  
[TURNTABLE](#) =289, [STRAIGHT](#) =290, [CAR](#) =291, [JOINT](#) =292,  
[NOTE](#) =293, [TEXT](#) =294, [MAIN](#) =295, [B](#) =296,  
[J](#) =297, [D](#) =298, [L](#) =299, [M](#) =300,  
[F](#) =301, [T](#) =302, [E](#) =303, [G](#) =304,  
[A](#) =305, [P](#) =306, [S](#) =307, [C](#) =308,  
[X](#) =309, [Y](#) =310, [Z](#) =311, [Q](#) =312,  
[W](#) =313, [H](#) =314, [BLOCK](#) =315, [TRK](#) =316,  
[SWITCHMOTOR](#) =317, [SIGNAL](#) =318, [ASPECT](#) =319, [SENSOR](#) =320,  
[CONTROL](#) =321, [ADJUSTABLE](#) =322, [PIER](#) =323 }

## Public Member Functions

- int [YY\\_MRRXtrkCad\\_PARSE](#) ([YY\\_MRRXtrkCad\\_PARSE\\_PARAM](#))
- virtual void [YY\\_MRRXtrkCad\\_ERROR](#) (char \*msg) [YY\\_MRRXtrkCad\\_ERROR\\_BODY](#)
- virtual int [YY\\_MRRXtrkCad\\_LEX](#) () [YY\\_MRRXtrkCad\\_LEX\\_BODY](#)
- [YY\\_MRRXtrkCad\\_CLASS](#) ([YY\\_MRRXtrkCad\\_CONSTRUCTOR\\_PARAM](#))
- virtual [~MRRXtrkCad](#) ()

## Public Attributes

- [YY\\_MRRXtrkCad\\_STYPE](#) [YY\\_MRRXtrkCad\\_LVAL](#)
- [YY\\_MRRXtrkCad\\_LTYPE](#) [YY\\_MRRXtrkCad\\_LLOC](#)
- int [YY\\_MRRXtrkCad\\_NERRS](#)
- int [YY\\_MRRXtrkCad\\_CHAR](#)
- int [YY\\_MRRXtrkCad\\_DEBUG\\_FLAG](#)

## Private Member Functions

- int [lookup\\_word](#) (const char \*word) const
- void [yyerror1](#) (const char \*message, const char \*s) const

## Private Attributes

- bool [scanEol](#)
- bool [scanToEND](#)
- int [fieldflag](#)
- double [CurrentScale](#)

## 9.193.1 Member Enumeration Documentation

### 9.193.1.1 YY\_MRRXtrkCad\_ENUM\_TOKEN

```
enum YY_MRRXtrkCad_INHERIT:YY_MRRXtrkCad_ENUM_TOKEN
```

#### Enumerator

YY_MRRXtrkCad_NULL_TOKEN	
INTEGER	
FLOAT	
STRING	
RESTOFLINE	
MULTILINE	
EOL	
UNTERMSTRING	
NOTWORD	
ENDSEGS	
ENDSIGNAL	
ENDBLOCK	
ENDTRACKS	
_VERSION	
TITLE	
MAPSCALE	

## Enumerator

ROOMSIZE	
SCALE	
HO	
N	
O	
LAYERS	
CURRENT	
STRUCTURE	
DRAW	
BEZIER	
BZRLIN	
CORNU	
SUBSEGS	
SUBSEND	
CURVE	
TURNOUT	
TURNTABLE	
STRAIGHT	
CAR	
JOINT	
NOTE	
TEXT	
MAIN	
B	
J	
D	
L	
M	
F	
T	
E	
G	
A	
P	
S	
C	
X	
Y	
Z	
Q	
W	
H	
BLOCK	
TRK	
SWITCHMOTOR	
SIGNAL	
ASPECT	

**Enumerator**

SENSOR	
CONTROL	
ADJUSTABLE	
PIER	

**9.193.2 Constructor & Destructor Documentation****9.193.2.1 ~MRRXtrkCad()**

```
virtual YY_MRRXtrkCad_INHERIT::~~MRRXtrkCad ( ) [inline], [virtual]
```

**9.193.3 Member Function Documentation****9.193.3.1 lookup\_word()**

```
int YY_MRRXtrkCad_INHERIT::lookup_word (
    const char * word ) const [private]
```

**9.193.3.2 YY\_MRRXtrkCad\_CLASS()**

```
YY_MRRXtrkCad_INHERIT::YY_MRRXtrkCad_CLASS (
    YY_MRRXtrkCad_CONSTRUCTOR_PARAM )
```

**9.193.3.3 YY\_MRRXtrkCad\_ERROR()**

```
virtual void YY_MRRXtrkCad_INHERIT::YY_MRRXtrkCad_ERROR (
    char * msg ) [virtual]
```



#### 9.193.3.4 YY\_MRRXtrkCad\_LEX()

```
virtual int YY_MRRXtrkCad_INHERIT::YY_MRRXtrkCad_LEX ( ) [virtual]
```

#### 9.193.3.5 YY\_MRRXtrkCad\_PARSE()

```
int YY_MRRXtrkCad_INHERIT::YY_MRRXtrkCad_PARSE (
    YY_MRRXtrkCad_PARSE_PARAM )
```

#### 9.193.3.6 yyerror1()

```
void YY_MRRXtrkCad_INHERIT::yyerror1 (
    const char * message,
    const char * s ) const [private]
```

### 9.193.4 Member Data Documentation

#### 9.193.4.1 CurrentScale

```
double YY_MRRXtrkCad_INHERIT::CurrentScale [private]
```

#### 9.193.4.2 fieldflag

```
int YY_MRRXtrkCad_INHERIT::fieldflag [private]
```

#### 9.193.4.3 scanEol

```
bool YY_MRRXtrkCad_INHERIT::scanEol [private]
```

#### 9.193.4.4 scanToEND

```
bool YY_MRRXtrkCad_INHERIT::scanToEND [private]
```

#### 9.193.4.5 YY\_MRRXtrkCad\_CHAR

```
int YY_MRRXtrkCad_INHERIT::YY_MRRXtrkCad_CHAR
```

#### 9.193.4.6 YY\_MRRXtrkCad\_DEBUG\_FLAG

```
int YY_MRRXtrkCad_INHERIT::YY_MRRXtrkCad_DEBUG_FLAG
```

#### 9.193.4.7 YY\_MRRXtrkCad\_LLOC

```
YY_MRRXtrkCad_LTYPE YY_MRRXtrkCad_INHERIT::YY_MRRXtrkCad_LLOC
```

#### 9.193.4.8 YY\_MRRXtrkCad\_LVAL

```
YY_MRRXtrkCad_STYPE YY_MRRXtrkCad_INHERIT::YY_MRRXtrkCad_LVAL
```

#### 9.193.4.9 YY\_MRRXtrkCad\_NERRS

```
int YY_MRRXtrkCad_INHERIT::YY_MRRXtrkCad_NERRS
```

### 9.194 yy\_MRRXtrkCad\_stype Union Reference

```
#include <MRRXtrkCad.tab.h>
```

## Public Attributes

- int [ival](#)
- char \* [sval](#)
- float [fval](#)
- [TrackBody](#) \* [tb](#)
- [TrackBodyElt](#) \* [tbe](#)
- [TurnoutBody](#) \* [trb](#)
- [TurnoutBodyElt](#) \* [trbe](#)
- [BezierBody](#) \* [tbb](#)
- [BezierBodyElt](#) \* [tbbe](#)
- [CornuBody](#) \* [tcb](#)
- [CornuBodyElt](#) \* [tcbe](#)
- [IntegerList](#) \* [il](#)
- [StringPairList](#) \* [spl](#)

### 9.194.1 Member Data Documentation

#### 9.194.1.1 fval

```
float yy_MRRXtrkCad_stype::fval
```

#### 9.194.1.2 il

```
IntegerList* yy_MRRXtrkCad_stype::il
```

#### 9.194.1.3 ival

```
int yy_MRRXtrkCad_stype::ival
```

#### 9.194.1.4 spl

```
StringPairList* yy_MRRXtrkCad_stype::spl
```

**9.194.1.5 sval**

`char* yy_MRRXtrkCad_stype::sval`

**9.194.1.6 tb**

`TrackBody* yy_MRRXtrkCad_stype::tb`

**9.194.1.7 tbb**

`BezierBody* yy_MRRXtrkCad_stype::tbb`

**9.194.1.8 tbbe**

`BezierBodyElt* yy_MRRXtrkCad_stype::tbbe`

**9.194.1.9 tbe**

`TrackBodyElt* yy_MRRXtrkCad_stype::tbe`

**9.194.1.10 tcb**

`CornuBody* yy_MRRXtrkCad_stype::tcb`

**9.194.1.11 tcbe**

`CornuBodyElt* yy_MRRXtrkCad_stype::tcbe`

## 9.194.1.12 trb

```
TurnoutBody* yy_MRRXtrkCad_stype::trb
```

## 9.194.1.13 trbe

```
TurnoutBodyElt* yy_MRRXtrkCad_stype::trbe
```

## 9.195 yyltype Struct Reference

```
#include <MRRXtrkCad.tab.h>
```

### Public Attributes

- int `timestamp`
- int `first_line`
- int `first_column`
- int `last_line`
- int `last_column`
- char \* `text`

### 9.195.1 Member Data Documentation

## 9.195.1.1 first\_column

```
int yytype::first_column
```

## 9.195.1.2 first\_line

```
int yytype::first_line
```

**9.195.1.3 last\_column**

```
int yytype::last_column
```

**9.195.1.4 last\_line**

```
int yytype::last_line
```

**9.195.1.5 text**

```
char* yytype::text
```

**9.195.1.6 timestamp**

```
int yytype::timestamp
```

## Chapter 10

# File Documentation

### 10.1 C++/Azatrax/Azatrax.h File Reference

```
#include "config.h"
```

#### Classes

- class [azatrax::Azatrax](#)  
*Azatrax I/O Class.*
- struct [azatrax::Azatrax::StateDataPacket](#)  
*Raw USB Data Packet.*

#### Namespaces

- [azatrax](#)  
*Azatrax C++ LibUSB 1.0 Interface.*

#### Macros

- #define [ErrorCode](#) int
- #define [stopwatchFract](#) status3
- #define [stopwatchSeconds](#) status4

#### 10.1.1 Macro Definition Documentation

#### 10.1.1.1 ErrorCode

```
#define ErrorCode int
```

Referenced by `azatrax::Azatrax::Identify_1()`, and `azatrax::Azatrax::ProductIdCode()`.

#### 10.1.1.2 stopwatchFract

```
#define stopwatchFract status3
```

#### 10.1.1.3 stopwatchSeconds

```
#define stopwatchSeconds status4
```

## 10.2 C++/Azatrax/mrd.h File Reference

```
#include <Azatrax.h>
```

### Classes

- class [azatrax::MRD](#)  
*MRD I/O Class.*
- union [azatrax::MRD::status1\\_union](#)  
*Status byte 1 union type.*
- union [azatrax::MRD::status2\\_union](#)  
*Status byte 2 union type.*

### Namespaces

- [azatrax](#)  
*Azatrax C++ LibUSB 1.0 Interface.*

## 10.3 C++/Azatrax/sl2.h File Reference

```
#include <Azatrax.h>
```



## Classes

- class [azatrax::SL2](#)  
*SL2 I/O Class.*
- union [azatrax::SL2::status1\\_union](#)  
*Status byte 1 union type (Output states)*
- union [azatrax::SL2::status2\\_union](#)  
*Status byte 2 union type (Input sense)*
- union [azatrax::SL2::status3\\_union](#)  
*Status byte 3 union type (Input control state)*

## Namespaces

- [azatrax](#)  
*Azatrax C++ LibUSB 1.0 Interface.*

## 10.4 C++/Azatrax/sr4.h File Reference

```
#include <Azatrax.h>
```

## Classes

- class [azatrax::SR4](#)  
*SR4 I/O Class.*
- union [azatrax::SR4::status1\\_union](#)  
*Status byte 1 union type (Outputs)*
- union [azatrax::SR4::status2\\_union](#)  
*Status byte 2 union type (Input sense)*
- union [azatrax::SR4::status3\\_union](#)  
*Status byte 3 union type (Input Control Status)*

## Namespaces

- [azatrax](#)  
*Azatrax C++ LibUSB 1.0 Interface.*

## 10.5 C++/FCFSupport/CallBack.h File Reference

```
#include <Common.h>
```

## Classes

- class [FCFSupport::WorkInProgressCallback](#)  
*Work In Progress Callback.*
- class [FCFSupport::LogMessageCallback](#)  
*A callback to log a message.*
- class [FCFSupport::ShowBannerCallback](#)  
*Display a page heading type message on the screen.*
- class [FCFSupport::TrainDisplayCallback](#)  
*Callback to manage a train status display.*
- class [FCFSupport::PauseCallback](#)  
*The Pause callback.*

## Namespaces

- [FCFSupport](#)  
*Namespace to hold all of the FCF Support code.*

## 10.6 C++/FCFSupport/Car.h File Reference

```
#include <Common.h>
```

## Classes

- class [FCFSupport::Car](#)  
*This class holds all of the information for a single car.*

## Namespaces

- [FCFSupport](#)  
*Namespace to hold all of the FCF Support code.*

## 10.7 C++/FCFSupport/CarType.h File Reference

```
#include <Common.h>
```

## Classes

- class [FCFSupport::CarType](#)  
*The [CarType](#) class represents a type of railroad car (rolling stock).*
- class [FCFSupport::CarGroup](#)  
*[Car](#) group class.*

## Namespaces

- [FCFSupport](#)  
*Namespace to hold all of the FCF Support code.*

## Typedefs

- typedef vector< char > [FCFSupport::CarTypeOrderVector](#)  
*A vector of ordered car types.*
- typedef map< char, CarType \*, less< char > > [FCFSupport::CarTypeMap](#)  
*A map of car types indexed by type character.*

## 10.8 C++/FCFSupport/Division.h File Reference

```
#include <Common.h>
#include <Station.h>
```

## Classes

- class [FCFSupport::Division](#)  
*The [Division](#) class implements a single division, which contains a number of contiguous stations.*

## Namespaces

- [FCFSupport](#)  
*Namespace to hold all of the FCF Support code.*

## Typedefs

- typedef vector< Division \* > [FCFSupport::DivisionVector](#)  
*A vector of divisions.*
- typedef map< int, Division \*, less< int > > [FCFSupport::DivisionMap](#)  
*A map of divisions, by integer index (division index).*
- typedef map< char, Division \*, less< char > > [FCFSupport::DivisionSymbolMap](#)  
*A map of divisions, by division symbol (a character).*

## 10.9 C++/FCFSupport/FCFSupportGroup.h File Reference

## 10.10 C++/FCFSupport/Industry.h File Reference

```
#include <Common.h>
#include <Station.h>
#include <Division.h>
#include <limits.h>
#include <iostream>
#include <assert.h>
```

### Classes

- class [FCFSupport::Industry](#)  
*The [Industry](#) class represents an industry.*

### Namespaces

- [FCFSupport](#)  
*Namespace to hold all of the FCF Support code.*

### Typedefs

- typedef vector< Car \* > [FCFSupport::CarVector](#)  
*A vector of cars.*
- typedef map< int, Industry \*, less< int > > [FCFSupport::IndustryMap](#)  
*A map of industry pointers indexed by an integer.*
- typedef vector< Industry \* > [FCFSupport::IndustryVector](#)  
*A vector of industry pointers.*

## 10.11 C++/FCFSupport/LQ24Printer.h File Reference

```
#include <Printer.h>
```

### Classes

- class [FCFSupport::LQ24PrinterDevice](#)  
*Class for an LQ24 compatible printer.*

## Namespaces

- [FCFSupport](#)

*Namespace to hold all of the FCF Support code.*

## 10.12 C++/FCFSupport/Owner.h File Reference

```
#include <Common.h>
```

## Classes

- class [FCFSupport::Owner](#)

*The [Owner](#) class describes a car owner.*

## Namespaces

- [FCFSupport](#)

*Namespace to hold all of the FCF Support code.*

## Typedefs

- typedef map< string, Owner \*, less< string > > [FCFSupport::OwnerMap](#)

*Map of owners, indexed by their initials.*

## 10.13 C++/FCFSupport/PathName.h File Reference

```
#include <Common.h>
```

## Classes

- class [FCFSupport::PathName](#)

*A Class that portably represents a pathname.*

## Namespaces

- [FCFSupport](#)

*Namespace to hold all of the FCF Support code.*

## Typedefs

- typedef vector< string > [FCFSupport::stringVector](#)  
*A Vector of strings.*

## 10.14 C++/TTSupport/PathName.h File Reference

```
#include <Common.h>
```

## Classes

- class [TTSupport::PathName](#)  
*A Class that portably represents a pathname.*

## Namespaces

- [TTSupport](#)  
*Time Table Support Namespace.*

## Typedefs

- typedef vector< string > [TTSupport::stringVector](#)  
*A Vector of strings.*

## 10.15 C++/FCFSupport/PDFPrinter.h File Reference

```
#include <Printer.h>  
#include <PDFPrinterSupport.h>  
#include <map>
```

## Classes

- class [FCFSupport::PDFPrinterDevice](#)  
*PDF Printer device.*

## Namespaces

- [FCFSupport](#)  
*Namespace to hold all of the FCF Support code.*

## Macros

- `#define oneColumnWidthFraction (((double)horizontalScaling)/100.0)`

### 10.15.1 Macro Definition Documentation

#### 10.15.1.1 oneColumnWidthFraction

```
#define oneColumnWidthFraction (((double)horizontalScaling)/100.0)
```

## 10.16 C++/FCFSupport/PDFPrinterSupport.h File Reference

```
#include <assert.h>
#include <time.h>
#include <PDFPrinterSupport.h>
#include <Common.h>
#include <iostream>
#include <sstream>
#include <map>
#include <vector>
```

## Classes

- class [FCFSupport::PDFFileStructures::CrossReferenceTable](#)  
*The cross reference table object.*
- class [FCFSupport::PDFFileStructures::IndirectObject](#)  
*Indirect object base class.*
- class [FCFSupport::PDFFileStructures::FreedObject](#)  
*A deleted indirect object.*
- class [FCFSupport::PDFFileStructures::Dictionary](#)  
*PDF Dictionary class.*
- class [FCFSupport::PDFFileStructures::PDFNameArray](#)  
*PDF Name array.*
- class [FCFSupport::PDFFileStructures::TypedDictionary](#)  
*Typed dictionary.*
- class [FCFSupport::PDFFileStructures::IndirectObjectDictionary](#)  
*PDF Indirect Object [Dictionary](#), used for named resources in a Resource [Dictionary](#).*
- class [FCFSupport::PDFFileStructures::ResourceDictionary](#)  
*Resource dictionary.*
- class [FCFSupport::PDFFileStructures::Rectangle](#)  
*A rectangle object.*

- class [FCFSupport::PDFFileStructures::PDFStream](#)  
*Stream object.*
- class [FCFSupport::PDFFileStructures::Page](#)  
*Describes a single page.*
- class [FCFSupport::PDFFileStructures::PageTree](#)  
*A tree of pages.*
- class [FCFSupport::PDFFileStructures::PageLabelDictionary](#)  
*Page label dictionary.*
- class [FCFSupport::PDFFileStructures::PageLabelTree](#)  
*A tree of page label dictionaries.*
- class [FCFSupport::PDFFileStructures::FontDictionary](#)  
*A Font dictionary object.*
- class [FCFSupport::PDFFileStructures::IndirectFloatVector](#)  
*Indirect array of floats.*
- class [FCFSupport::PDFFileStructures::Type1FontDictionary](#)  
*Type 1 Font dictionary.*
- class [FCFSupport::PDFFileStructures::PostScriptStandardType1FontDictionary](#)  
*A standard Type1 PostScript font dictionary.*
- class [FCFSupport::PDFFileStructures::CatalogDictionary](#)  
*Master catalog of the PDF file.*
- class [FCFSupport::PDFFileStructures::InformationDirectory](#)  
*Information directory.*

## Namespaces

- [FCFSupport](#)  
*Namespace to hold all of the FCF Support code.*
- [FCFSupport::PDFFileStructures](#)  
*PDF File support structures.*

## Typedefs

- typedef map< string, IndirectObject \*, less< string > > [FCFSupport::PDFFileStructures::NamedIndirectObjectMap](#)  
*A "vector" of named indirect objects, implemented as a map.*
- typedef vector< PDFStream \* > [FCFSupport::PDFFileStructures::PDFStreamVector](#)  
*A vector of PDF Streams.*
- typedef vector< PageLabelTree \* > [FCFSupport::PDFFileStructures::PageLabelTreeKidVector](#)  
*Map of [PageLabelTree](#) kids.*
- typedef map< int, PageLabelDictionary \*, less< int > > [FCFSupport::PDFFileStructures::PageLabelDictionaryNumMap](#)  
*Map of [PageLabelDictionary](#) numbers.*



## Functions

- struct tm \* [localtime\\_r](#) (const time\_t \*, struct tm \*)
- char \* [asctime\\_r](#) (const struct tm \*, char \*)
- string [FCFSupport::PDFFileStructures::QuotePDFString](#) (const string &str)  
*Quote a string (protect special character with a backslash).*
- ostream & [FCFSupport::operator<<](#) (ostream &stream, const PDFFileStructures::PDFNameArray &pnarray)  
*Output stream operator for PDFNameArrays.*

### 10.16.1 Function Documentation

#### 10.16.1.1 asctime\_r()

```
char* asctime_r (
    const struct tm * ,
    char * )
```

#### 10.16.1.2 localtime\_r()

```
struct tm* localtime_r (
    const time_t * ,
    struct tm * )
```

## 10.17 C++/FCFSupport/PostScriptPrinter.h File Reference

```
#include <Printer.h>
```

## Classes

- class [FCFSupport::PostScriptPrinterDevice](#)  
*Derived class for printing on Postscript printers.*

## Namespaces

- [FCFSupport](#)  
*Namespace to hold all of the FCF Support code.*

## 10.18 C++/FCFSupport/Printer.h File Reference

```
#include <Common.h>
#include <iostream>
#include <fstream>
#include <stdio.h>
```

### Classes

- class [FCFSupport::PrinterDevice](#)  
*Base class for printer devices (hard copy output).*

### Namespaces

- [FCFSupport](#)  
*Namespace to hold all of the FCF Support code.*

## 10.19 C++/FCFSupport/Station.h File Reference

```
#include <Common.h>
```

### Classes

- class [FCFSupport::Station](#)  
*The [Station](#) class implements a single station.*

### Namespaces

- [FCFSupport](#)  
*Namespace to hold all of the FCF Support code.*

### Typedefs

- typedef vector< Station \* > [FCFSupport::StationVector](#)  
*A station vector.*
- typedef map< int, Station \*, less< int > > [FCFSupport::StationMap](#)  
*A station map by integer index.*

## 10.20 C++/TTSupport/Station.h File Reference

```
#include <Common.h>
#include <iostream>
#include <fstream>
```

### Classes

- class [TTSupport::Occupied](#)  
*This class records a train sitting on a storage track during a specified time frame.*
- class [TTSupport::TimeRange](#)  
*The [TimeRange](#) class implements a range of times.*
- class [TTSupport::StorageTrack](#)  
*The [StorageTrack](#) class implements a storage track.*
- class [TTSupport::Station](#)  
*The [Station](#) class implements a station.*

### Namespaces

- [TTSupport](#)  
*Time Table Support Namespace.*

### Typedefs

- typedef map< TimeRange, Occupied, less< TimeRange > > [TTSupport::OccupiedMap](#)  
*The [Occupied](#) Map type, ordered by time ranges.*
- typedef map< string, StorageTrack, less< string > > [TTSupport::StorageTrackMap](#)  
*Storage track map.*
- typedef vector< Station > [TTSupport::StationVector](#)  
*[Station](#) Vector.*

## 10.21 C++/FCFSupport/SwitchList.h File Reference

```
#include <iostream>
#include <Common.h>
#include <Train.h>
#include <Industry.h>
#include <Car.h>
#include <Station.h>
```

## Classes

- class [FCFSupport::SwitchListElement](#)  
*This class implements each switch list element.*
- union [FCFSupport::SwitchListElement::StationOrIndustry](#)  
*A const pointer to a train's stop, which can be either a station or an industry, depending on what kind of train it is.*
- class [FCFSupport::SwitchList](#)  
*The global switch list structure.*

## Namespaces

- [FCFSupport](#)  
*Namespace to hold all of the FCF Support code.*

## Typedefs

- typedef vector< SwitchListElement > [FCFSupport::SwitchListElementVector](#)  
*A vector of switch list elements.*

## Functions

- ostream & [FCFSupport::operator<<](#) (ostream &stream, const SwitchListElement &element)  
*Output stream operator for SwitchListElements.*

## 10.22 C++/FCFSupport/System.h File Reference

```
#include <Common.h>
#include <PathName.h>
#include <Station.h>
#include <Division.h>
#include <Train.h>
#include <Industry.h>
#include <CarType.h>
#include <Owner.h>
#include <Car.h>
#include <CallBack.h>
#include <Printer.h>
#include <SwitchList.h>
```

## Classes

- class [FCFSupport::System](#)  
*This is the main Freight [Car](#) Forwarder class.*

## Namespaces

- [FCFSupport](#)

*Namespace to hold all of the FCF Support code.*

## 10.23 C++/FCFSupport/TextPrinter.h File Reference

```
#include <Printer.h>
```

## Classes

- class [FCFSupport::TextPrinterDevice](#)

*Derived class for printing on generic plain text printers.*

## Namespaces

- [FCFSupport](#)

*Namespace to hold all of the FCF Support code.*

## 10.24 C++/FCFSupport/Train.h File Reference

```
#include <Common.h>
#include <Station.h>
#include <Division.h>
#include <Industry.h>
```

## Classes

- class [FCFSupport::Train](#)  
*The [Train](#) class represents a train.*
- union [FCFSupport::Train::StationOrIndustry](#)

*Union of stations or industries, used for stops.*

## Namespaces

- [FCFSupport](#)

*Namespace to hold all of the FCF Support code.*

## Typedefs

- typedef map< int, Train \*, less< int > > [FCFSupport::TrainMap](#)  
*A map of trains, indexed by integer (train index).*
- typedef map< string, Train \*, less< string > > [FCFSupport::TrainNameMap](#)  
*A map of trains, indexed by string ([Train](#) name).*

## 10.25 C++/TTSupport/Train.h File Reference

```
#include <Common.h>
#include <Cab.h>
```

## Classes

- class [TTSupport::Stop](#)  
*This class implements a stop.*
- class [TTSupport::Train](#)  
*This class implements a train.*

## Namespaces

- [TTSupport](#)  
*Time Table Support Namespace.*

## Typedefs

- typedef vector< Stop > [TTSupport::StopVector](#)  
*A vector of stops.*
- typedef map< string, Train \*, less< string > > [TTSupport::TrainNumberMap](#)  
*[Train](#) number map, indexed by train number (symbol).*

## 10.26 C++/ParserClasses/BezierBody.h File Reference

```
#include <TrackBody.h>
#include <string.h>
#include <iostream>
```

## Classes

- class [Parsers::BezierBodyElt](#)  
*Bezier Body elements: T, E, S, and C lines are collected.*
- struct [Parsers::BezierBodyElt::Pos](#)  
*Position structure.*
- class [Parsers::BezierBody](#)  
*List of Bezier body lines (T, E, S, and C lines).*

## Namespaces

- [Parsers](#)  
*File-based parser classes.*

## Macros

- `#define angle radius`
- `#define len0 ang0`
- `#define len1 ang1`

### 10.26.1 Macro Definition Documentation

#### 10.26.1.1 [angle](#)

```
#define angle radius
```

#### 10.26.1.2 [len0](#)

```
#define len0 ang0
```

#### 10.26.1.3 [len1](#)

```
#define len1 ang1
```

## 10.27 C++/ParserClasses/CornuBody.h File Reference

```
#include <TrackBody.h>
#include <string.h>
#include <iostream>
```

### Classes

- class [Parsers::CornuBodyElt](#)  
*Cornu Body elements: T, E, S, and C lines are collected.*
- struct [Parsers::CornuBodyElt::Pos](#)  
*Position structure.*
- class [Parsers::CornuBody](#)  
*List of Cornu body lines (T, E, S, and C lines).*

### Namespaces

- [Parsers](#)  
*File-based parser classes.*

### Macros

- `#define` [angle](#) radius
- `#define` [len0](#) ang0
- `#define` [len1](#) ang1

#### 10.27.1 Macro Definition Documentation

##### 10.27.1.1 angle

```
#define angle radius
```

##### 10.27.1.2 len0

```
#define len0 ang0
```



## 10.27.1.3 len1

```
#define len1 angl
```

## 10.28 C++/ParserClasses/IntegerList.h File Reference

### Classes

- class [Parsers::IntegerList](#)

The [IntegerList](#) class implements a linked list of integers, used for turnout route lists.

### Namespaces

- [Parsers](#)

File-based parser classes.

## 10.29 C++/ParserClasses/MRRXtrkCad.tab.h File Reference

```
#include "config.h"
#include <stdio.h>
#include <iostream>
#include <ctype.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <ParseFile.h>
#include <IntegerList.h>
#include <StringPairList.h>
```

### Classes

- union [yy\\_MRRXtrkCad\\_stype](#)
- struct [yyltype](#)
- class [YY\\_MRRXtrkCad\\_INHERIT](#)

## Macros

- `#define YY_USE_CLASS`
- `#define TRUE true`
- `#define FALSE false`
- `#define RADIANS(x) (((x) / 180.0) * M_PI)`
- `#define YY_MRRXtrkCad_CLASS MRRXtrkCad`
- `#define YY_MRRXtrkCad_INHERIT : public LayoutFile`
- `#define YY_MRRXtrkCad_CONSTRUCTOR_PARAM const char * filename`
- `#define YY_MRRXtrkCad_CONSTRUCTOR_INIT : LayoutFile (filename,this)`
- `#define YY_MRRXtrkCad_CONSTRUCTOR_CODE`
- `#define YY_MRRXtrkCad_MEMBERS`
- `#define YY_MRRXtrkCad_LSP_NEEDED 1`
- `#define YY_MRRXtrkCad_ERROR_VERBOSE`
- `#define YY_MRRXtrkCad_DEBUG 1`
- `#define YY_MRRXtrkCad_STYPE yy_MRRXtrkCad_stype`
- `#define YY_MRRXtrkCad_COMPATIBILITY 0`
- `#define YY_MRRXtrkCad_USE_GOTO 0`
- `#define BISON_YYLTYPE_ISDECLARED`
- `#define YY_MRRXtrkCad_LTYPE yyltype`
- `#define YY_MRRXtrkCad_PARSE yyparse`
- `#define YY_MRRXtrkCad_LEX yylex`
- `#define YY_MRRXtrkCad_LVAL yylval`
- `#define YY_MRRXtrkCad_LLOC yyloc`
- `#define YY_MRRXtrkCad_CHAR yychar`
- `#define YY_MRRXtrkCad_NERRS yynerrs`
- `#define YY_MRRXtrkCad_DEBUG_FLAG yydebug`
- `#define YY_MRRXtrkCad_ERROR yyerror`
- `#define YY_MRRXtrkCad_PARSE_PARAM void`
- `#define YY_MRRXtrkCad_LEX_BODY`
- `#define YY_MRRXtrkCad_ERROR_BODY`
- `#define YY_MRRXtrkCad_USE_CONST_TOKEN 0`

## Typedefs

- `typedef struct yyltype yyltype`

## Variables

- `const double INCHESperMM`
- `const double FEETperMM`
- `const double YARDSperMM`
- `const double METERSperMM`
- `const double CENTIMETERSperMM`
- `const double HOScale`
- `const double NScale`
- `const double OScale`
- `const double IScale`
- `const double GScale`

## 10.29.1 Macro Definition Documentation

### 10.29.1.1 BISON\_YYLTYPE\_ISDECLARED

```
#define BISON_YYLTYPE_ISDECLARED
```

### 10.29.1.2 FALSE

```
#define FALSE false
```

### 10.29.1.3 RADIANS

```
#define RADIANS(  
    x ) ((x) / 180.0) * M_PI
```

### 10.29.1.4 TRUE

```
#define TRUE true
```

### 10.29.1.5 YY\_MRRXtrkCad\_CHAR

```
#define YY_MRRXtrkCad_CHAR yychar
```

### 10.29.1.6 YY\_MRRXtrkCad\_CLASS

```
#define YY_MRRXtrkCad_CLASS MRRXtrkCad
```

**10.29.1.7 YY\_MRRXtrkCad\_COMPATIBILITY**

```
#define YY_MRRXtrkCad_COMPATIBILITY 0
```

**10.29.1.8 YY\_MRRXtrkCad\_CONSTRUCTOR\_CODE**

```
#define YY_MRRXtrkCad_CONSTRUCTOR_CODE
```

**Value:**

```
CurrentScale = 1.0;\n\n    scanEol = false; \n    scanToEND = false;\n    /*YY_MRRXtrkCad_DEBUG_FLAG = 1;*/
```

**10.29.1.9 YY\_MRRXtrkCad\_CONSTRUCTOR\_INIT**

```
#define YY_MRRXtrkCad_CONSTRUCTOR_INIT : LayoutFile (filename,this)
```

**10.29.1.10 YY\_MRRXtrkCad\_CONSTRUCTOR\_PARAM**

```
#define YY_MRRXtrkCad_CONSTRUCTOR_PARAM const char * filename
```

**10.29.1.11 YY\_MRRXtrkCad\_DEBUG**

```
#define YY_MRRXtrkCad_DEBUG 1
```

**10.29.1.12 YY\_MRRXtrkCad\_DEBUG\_FLAG**

```
#define YY_MRRXtrkCad_DEBUG_FLAG yydebug
```

**10.29.1.13 YY\_MRRXtrkCad\_ERROR**

```
#define YY_MRRXtrkCad_ERROR yyerror
```

**10.29.1.14 YY\_MRRXtrkCad\_ERROR\_BODY**

```
#define YY_MRRXtrkCad_ERROR_BODY
```

**10.29.1.15 YY\_MRRXtrkCad\_ERROR\_VERBOSE**

```
#define YY_MRRXtrkCad_ERROR_VERBOSE
```

**10.29.1.16 YY\_MRRXtrkCad\_INHERIT**

```
#define YY_MRRXtrkCad_INHERIT : public LayoutFile
```

**10.29.1.17 YY\_MRRXtrkCad\_LEX**

```
#define YY_MRRXtrkCad_LEX yylex
```

**10.29.1.18 YY\_MRRXtrkCad\_LEX\_BODY**

```
#define YY_MRRXtrkCad_LEX_BODY
```

**10.29.1.19 YY\_MRRXtrkCad\_LLOC**

```
#define YY_MRRXtrkCad_LLOC yylloc
```

**10.29.1.20 YY\_MRRXtrkCad\_LSP\_NEEDED**

```
#define YY_MRRXtrkCad_LSP_NEEDED 1
```

**10.29.1.21 YY\_MRRXtrkCad\_LTYPE**

```
#define YY_MRRXtrkCad_LTYPE yyltype
```

**10.29.1.22 YY\_MRRXtrkCad\_LVAL**

```
#define YY_MRRXtrkCad_LVAL yylval
```

**10.29.1.23 YY\_MRRXtrkCad\_MEMBERS**

```
#define YY_MRRXtrkCad_MEMBERS
```

**Value:**

```
virtual ~MRRXtrkCad() {}\  
private\  
    int lookup_word(const char *word) const;\  
    void yyerror1(const char *message, const char *s) const;\  
    bool scanEol, scanToEnd; \  
    int fieldflag;\  
    double CurrentScale;
```

**10.29.1.24 YY\_MRRXtrkCad\_NERRS**

```
#define YY_MRRXtrkCad_NERRS yynerrs
```

**10.29.1.25 YY\_MRRXtrkCad\_PARSE**

```
#define YY_MRRXtrkCad_PARSE yyparse
```

#### 10.29.1.26 YY\_MRRXtrkCad\_PARSE\_PARAM

```
#define YY_MRRXtrkCad_PARSE_PARAM void
```

#### 10.29.1.27 YY\_MRRXtrkCad\_STYPE

```
#define YY_MRRXtrkCad_STYPE yy\_MRRXtrkCad\_stype
```

#### 10.29.1.28 YY\_MRRXtrkCad\_USE\_CONST\_TOKEN

```
#define YY_MRRXtrkCad_USE_CONST_TOKEN 0
```

#### 10.29.1.29 YY\_MRRXtrkCad\_USE\_GOTO

```
#define YY_MRRXtrkCad_USE_GOTO 0
```

#### 10.29.1.30 YY\_USE\_CLASS

```
#define YY_USE_CLASS
```

### 10.29.2 Typedef Documentation

#### 10.29.2.1 yyltype

```
typedef struct yyltype yyltype
```

### 10.29.3 Variable Documentation

**10.29.3.1 CENTIMETERSperMM**

```
const double CENTIMETERSperMM
```

**10.29.3.2 FEETperMM**

```
const double FEETperMM
```

**10.29.3.3 GScale**

```
const double GScale
```

**10.29.3.4 HOScale**

```
const double HOScale
```

**10.29.3.5 INCHESperMM**

```
const double INCHESperMM
```

**10.29.3.6 IScale**

```
const double IScale
```

**10.29.3.7 METERSperMM**

```
const double METERSperMM
```



### 10.29.3.8 NScale

```
const double NScale
```

### 10.29.3.9 OScale

```
const double OScale
```

### 10.29.3.10 YARDSperMM

```
const double YARDSperMM
```

## 10.30 C++/ParserClasses/ParseFile.h File Reference

```
#include <iostream>
#include <string.h>
#include <stdio.h>
#include <TrackGraph.h>
```

### Classes

- class [Parsers::ParseFile](#)  
*Virtual base class for file-based parsers.*
- class [Parsers::LayoutFile](#)  
*File to parse an XTrkCad layout file and create a track graph.*

### Namespaces

- [Parsers](#)  
*File-based parser classes.*

## 10.31 C++/ParserClasses/ParserClassesGroup.h File Reference

### Classes

- class [Parsers::MRRXtrkCad](#)  
*MRRXtrkCad parser class.*

## Namespaces

- [Parsers](#)

*File-based parser classes.*

## 10.32 C++/ParserClasses/SocketPair.h File Reference

### Functions

- list [tcl\\_socketpair](#) ()

*Tcl interface to socketpair.*

## 10.33 C++/ParserClasses/TrackBody.h File Reference

```
#include <iostream>
```

### Classes

- class [Parsers::TrackBodyElt](#)  
*Track endpoint elements (T and E lines).*
- class [Parsers::TrackBody](#)  
*List of track endpoints (T and E lines).*

## Namespaces

- [Parsers](#)

*File-based parser classes.*

## 10.34 C++/ParserClasses/TrackGraph.h File Reference

```
#include <iostream>
#include <boost/config.hpp>
#include <boost/version.hpp>
#include <boost/graph/adjacency_list.hpp>
#include <list>
#include <TrackBody.h>
#include <TurnoutBody.h>
#include <BezierBody.h>
#include <CornuBody.h>
#include <IntegerList.h>
#include <StringPairList.h>
```

## Classes

- struct [Parsers::SegPos](#)  
*Segment position, endpoint or other coordinate.*
- struct [Parsers::SegVector](#)  
*Segemnt structure.*
- struct [Parsers::TurnoutGraphic](#)  
*Structure holding a turnout's graphical information.*
- struct [Parsers::RouteVec](#)  
*Route structure.*
- struct [Parsers::TurnoutRoutelist](#)  
*Turnout route list structure.*
- class [Parsers::TrackGraph](#)  
*Track Graph class, which encapsulates the track graphs.*
- struct [Parsers::TrackGraph::EdgeValues](#)  
*Uncompressed graph edge values.*
- struct [Parsers::TrackGraph::NodeValues](#)  
*Uncompressed graph node values.*
- struct [Parsers::TrackGraph::CompressedEdgeValues](#)  
*Compressed graph edge values.*
- struct [Parsers::TrackGraph::Point](#)  
*Position structure.*
- struct [Parsers::TrackGraph::CompressedNodeValues](#)  
*Compressed graph node values.*
- class [Parsers::TrackGraph::Transform2D](#)  
*Two dimensional transform class.*

## Namespaces

- [Parsers](#)  
*File-based parser classes.*

## 10.35 C++/ParserClasses/TurnoutBody.h File Reference

```
#include <TrackBody.h>
#include <IntegerList.h>
#include <string.h>
#include <iostream>
```

## Classes

- class [Parsers::TurnoutBodyElt](#)  
*Turnout body elements: T, E, P, S, C, and J lines are collected.*
- struct [Parsers::TurnoutBodyElt::Pos](#)  
*Position structure.*
- class [Parsers::TurnoutBody](#)  
*List of turnout body lines (T, E, P, S, C, and J lines).*

## Namespaces

- [Parsers](#)

*File-based parser classes.*

## Macros

- `#define` [angle](#) radius
- `#define` [len0](#) ang0
- `#define` [len1](#) ang1

### 10.35.1 Macro Definition Documentation

#### 10.35.1.1 [angle](#)

```
#define angle radius
```

Referenced by `Parsers::TurnoutBodyElt::GetTurnoutJointSegment()`.

#### 10.35.1.2 [len0](#)

```
#define len0 ang0
```

Referenced by `Parsers::TurnoutBodyElt::GetTurnoutJointSegment()`.

#### 10.35.1.3 [len1](#)

```
#define len1 angl
```

Referenced by `Parsers::TurnoutBodyElt::GetTurnoutJointSegment()`.

## 10.36 C++/RailDriver/RaildriverIO.h File Reference

```
#include <hidapi.h>
```

## Classes

- class [RaildriverIO](#)  
*Low-level Raildriver I/O functions.*

## 10.37 C++/TclSocketCAN/TclSocketCAN.i File Reference

### Macros

- #define [SWIG\\_name](#) "Tclsocketcan"
- #define [SWIG\\_version](#) TCLSOCKETCAN\_VERSIONLIB

### Functions

- SWIGEXPORT int [Tclsocketcan\\_SafeInit](#) (Tcl\_Interp \*)
- int [SocketCAN](#) (Tcl\_Interp \*interp, const char \*candev)  
*Open a CAN Socket.*

### Variables

- module [TclSocketCAN](#)
- include typemaps [i](#)

### 10.37.1 Macro Definition Documentation

#### 10.37.1.1 [SWIG\\_name](#)

```
#define SWIG_name "Tclsocketcan"
```

#### 10.37.1.2 [SWIG\\_version](#)

```
#define SWIG_version TCLSOCKETCAN_VERSIONLIB
```

## 10.38 C++/TTSupport/Cab.h File Reference

```
#include <Common.h>  
#include <iostream>
```

## Classes

- class [TTSupport::Cab](#)  
*This class maintains information about cabs.*

## Namespaces

- [TTSupport](#)  
*Time Table Support Namespace.*

## Typedefs

- typedef map< string, Cab \*, less< string > > [TTSupport::CabNameMap](#)  
*[Cab](#) name map, cabs indexed by name.*

## 10.39 C++/TTSupport/TimeTableSystem.h File Reference

```
#include <string.h>
#include <Common.h>
#include <PathName.h>
#include <Station.h>
#include <Cab.h>
#include <Train.h>
#include <list>
#include <unordered_map>
```

## Classes

- struct [TTSupport::hash](#)  
*Option hash map, used for Print options.*
- struct [TTSupport::eqstr](#)
- class [TTSupport::StationTimes](#)  
*[Station](#) times class, used by the LaTeX generator methods.*
- class [TTSupport::TimeTableSystem](#)  
*This is the main Time Table Class.*

## Namespaces

- [TTSupport](#)  
*Time Table Support Namespace.*

## Macros

- `#define` [USE\\_UNORDERED\\_MAP](#)

## Typedefs

- `typedef vector< double >` [TTSupport::doubleVector](#)  
*A Vector of doubles.*
- `typedef std::unordered_map< const char *, std::string, hash, eqstr >` [TTSupport::OptionHashMap](#)
- `typedef list< Train * >` [TTSupport::TrainList](#)  
*List of trains.*
- `typedef map< string, StationTimes, less< string > >` [TTSupport::TrainStationTimes](#)  
*Map of station times, indexed by train number.*
- `typedef map< int, TrainStationTimes, less< int > >` [TTSupport::TrainTimesAtStation](#)  
*Map of maps of station times, indexed by station index.*
- `typedef list< string >` [TTSupport::StringList](#)  
*List of strings.*

## Functions

- `const char *` [TTSupport::StringListToString](#) (const StringList &list)  
*Convert a list of strings to a flat string.*
- `bool` [TTSupport::StringListFromString](#) (string strlinList, StringList &result)  
*Convert a flat string to a list of strings.*

### 10.39.1 Macro Definition Documentation

#### 10.39.1.1 USE\_UNORDERED\_MAP

```
#define USE_UNORDERED_MAP
```

## 10.40 C++/TTSupport/TimeTableSystemTcl.h File Reference

### Functions

- TimeTableSystem \* [NewCreateTimeTable](#) (const char \*name, int timescale, int timeinterval)  
*Tcl constructor to create a new TimeTable.*
- TimeTableSystem \* [OldCreateTimeTable](#) (const char \*filename, char \*\*outmessage)  
*Tcl constructor to create a time table system from an existing file.*
- int [ForEveryStation](#) (Tcl\_Interp \*interp, TimeTableSystem \*timetable, Tcl\_Obj \*variable, Tcl\_Obj \*body)  
*Tcl looping construct for Stations.*
- int [ForEveryCab](#) (Tcl\_Interp \*interp, TimeTableSystem \*timetable, Tcl\_Obj \*variable, Tcl\_Obj \*body)  
*Tcl looping construct for Cabs.*
- int [ForEveryTrain](#) (Tcl\_Interp \*interp, TimeTableSystem \*timetable, Tcl\_Obj \*variable, Tcl\_Obj \*body)  
*Tcl looping construct for Trains.*
- int [ForEveryNote](#) (Tcl\_Interp \*interp, TimeTableSystem \*timetable, Tcl\_Obj \*variable, Tcl\_Obj \*body)  
*Tcl looping construct for notes.*
- int [ForEveryPrintOption](#) (Tcl\_Interp \*interp, TimeTableSystem \*timetable, Tcl\_Obj \*variable, Tcl\_Obj \*body)  
*Tcl looping construct for print options.*
- int [TT\\_StringListToList](#) (Tcl\_Interp \*interp, const char \*stringList)  
*Tcl function to convert a serialized string list to a Tcl list.*
- int [TT\\_ListToStringListString](#) (Tcl\_Interp \*interp, Tcl\_Obj \*list)  
*Tcl function to convert a Tcl list to a serialized string list.*

### Variables

- apply int [Tcl\\_Result](#) { int [TTSupport::ForEveryStation](#) }

## 10.41 C++/TTSupport/TTSupportGroup.h File Reference

### Namespaces

- [TTSupport](#)  
*Time Table Support Namespace.*

## 10.42 C++/wiringPi/tclwiringpi.i File Reference

```
#include <wiringPi.h>
#include <wiringPiI2C.h>
#include <wiringPiSPI.h>
#include <mcp23008.h>
#include <mcp23017.h>
```



## Macros

- `#define SWIG_name "Tclwiringpi"`
- `#define SWIG_version "1.0.0"`

## Variables

- module `Tclwiringpi`

### 10.42.1 Macro Definition Documentation

#### 10.42.1.1 SWIG\_name

```
#define SWIG_name "Tclwiringpi"
```

#### 10.42.1.2 SWIG\_version

```
#define SWIG_version "1.0.0"
```

### 10.42.2 Variable Documentation

#### 10.42.2.1 Tclwiringpi

```
module Tclwiringpi
```

##### Initial value:

```
{
```

```
static const char rcsid[] = "@(#) : $Id$"
```

## 10.43 Doc/doxygen/titlepage.h File Reference

## 10.44 Scripts/CMri/cmri.tcl File Reference

### Classes

- class [cmri::CMri](#)  
*Main C/MRI interface class.*

### Namespaces

- [cmri](#)  
*CMR/I Tcl Serial Port Interface.*

### Typedefs

- typedef int [cmri::uatype](#)  
*Board address type.*
- typedef int [cmri::ubyte](#)  
*Unsigned byte.*
- typedef listtype [cmri::ByteList](#)  
*List of bytes.*

### Enumerations

- enum [cmri::CardType](#) { [cmri::USIC](#), [cmri::SUSIC](#), [cmri::SMINI](#) }  
*Card type codes.*

## 10.45 Scripts/Common/CabWidgets.tcl File Reference

### Classes

- class [CabWidgets::LocomotiveSpeed](#)  
*Locomotive Speed widget.*
- class [CabWidgets::LocomotiveDirection](#)  
*Locomotive Direction widget.*
- class [CabWidgets::SelectLocomotive](#)  
*Select or enter a Locomotive address.*

## Namespaces

- [CabWidgets](#)  
*Cab Widget code.*

## 10.46 Scripts/Common/CommonTclGroup.h File Reference

## 10.47 Scripts/Common/CTCPanel2.tcl File Reference

### Classes

- class [CTCPanel::CTCPanel](#)  
*Main CTC Panel megawidget.*
- class [CTCPanel::SWPlate](#)  
*Switch plate object type.*
- class [CTCPanel::SIGPlate](#)  
*Signal plate object type.*
- class [CTCPanel::CodeButton](#)  
*Code button object type.*
- class [CTCPanel::Toggle](#)  
*Toggle switch object type.*
- class [CTCPanel::Lamp](#)  
*Lamp object type.*
- class [CTCPanel::PushButton](#)  
*Push Button object type.*
- class [CTCPanel::CTCLabel](#)  
*CTC Label object type.*
- class [CTCPanel::SchLabel](#)  
*Schematic Label object type.*
- class [CTCPanel::Switch](#)  
*Switch (turnout) object type.*
- class [CTCPanel::Signal](#)  
*Signal object type.*
- class [CTCPanel::StraightBlock](#)  
*Straight Block object type.*
- class [CTCPanel::EndBumper](#)  
*End Bumper object type.*
- class [CTCPanel::CurvedBlock](#)  
*Curved Block object type.*
- class [CTCPanel::ScissorCrossover](#)  
*Scissor *Crossover* (turnout) object type.*
- class [CTCPanel::Crossover](#)  
**Crossover* (turnout) object type.*
- class [CTCPanel::Crossing](#)  
*Crossing object type.*

- class [CTCPanel::SingleSlip](#)  
*Single Slip (turnout) object type.*
- class [CTCPanel::DoubleSlip](#)  
*Double Slip (turnout) object type.*
- class [CTCPanel::ThreeWaySW](#)  
*Three Way Switch (turnout) object type.*
- class [CTCPanel::HiddenBlock](#)  
*Hidden Block object type.*
- class [CTCPanel::StubYard](#)  
*Stub Yard object type.*
- class [CTCPanel::ThroughYard](#)  
*Through Yard object type.*

## Namespaces

- [CTCPanel](#)  
*CTC Panel code, Version 2.*

## Functions

- [CTCPanel::leverMethods](#) (hasCenter)  
*Macro to add lever methods to object types.*
- [CTCPanel::verifyDoubleMethod](#) ()  
*Macro to add a verify double method to a snit type.*
- [CTCPanel::verifyBoolMethod](#) ()  
*Macro to add a verify boolean method to a snit type.*
- [CTCPanel::verifyColorMethod](#) ()  
*Macro to add a verify color method to a snit type.*
- [CTCPanel::verifyOrientation8Method](#) ()  
*Macro to add a verify 8-way orientation method to a snit type.*
- [CTCPanel::verifyPositionMethod](#) ()  
*Macro to add a verify position method to a snit type.*
- [CTCPanel::standardMethods](#) ()  
*Macro to add a standard set of methods to an object type.*
- [CTCPanel::trackworkmethods](#) ()  
*Macro to include trackwork drawing methods.*

## 10.48 Scripts/Common/fileentry.tcl File Reference

### Namespaces

- [FileEntry](#)  
*This is a specialized form of the LabelEntry widget intended for selecting file names.*

## Functions

- [FileEntry::create](#) (path,...)  
*Creation procedure.*
- [FileEntry::configure](#) (path,...)  
*Configuration procedure: configure one or more options for this widget.*
- [FileEntry::cget](#) (path, option)  
*Configuration option accessor procedure: access one option directly.*
- [FileEntry::bind](#) (path,...)  
*Bind function.*
- [FileEntry::\\_path\\_command](#) (path, cmd, larg)  
*Path command for this megawidget.*
- [FileEntry::\\_destroy](#) (path)  
*Destructor function.*
- [FileEntry::\\_openFile](#) (path)  
*Prodedure bound to the file open button.*

## 10.49 Scripts/Common/gettext.tcl File Reference

### Namespaces

- [gettext](#)  
*Localization functions.*

### Functions

- [gettext::\\_m](#) (msgid,...)  
*Handle messages with a context hint prefix (eg Label|lab).*
- [gettext::\\_mx](#) (...)  
*Get maxlength of a set of messages with a context hint prefix.*
- [gettext::\\_](#) (...)  
*Get a localized from the message catalog and deal with formating possible arguments, by calling ::msgcat::mc.*

## 10.50 Scripts/Common/HTMLHelp.tcl File Reference

### Classes

- class [HTMLHelp::HTMLHelp](#)  
*A widget that implements a help dialog that renders HTML coded help pages (generally generated from LaTeX using tex4ht's htlatex script).*

## Namespaces

- [HTMLHelp](#)

*[HTMLHelp](#) namespace, which contains the [HTMLHelp](#) snit widget adapter object and associated code.*

## 10.51 Scripts/Common/labelcombobox.tcl File Reference

### Namespaces

- [LabelComboBox](#)

*This is a specialized form of the [LabelFrame](#) widget containing a [ComboBox](#) Widget.*

### Functions

- [LabelComboBox::create](#) (path,...)  
*Procedure to create a [LabelComboBox](#).*
- [LabelComboBox::configure](#) (path,...)  
*Procedure to configure a [LabelComboBox](#).*
- [LabelComboBox::cget](#) (path, option)  
*Procedure to get a configuration option.*
- [LabelComboBox::bind](#) (path,...)  
*Procedure to set a binding on the [ComboBox](#) entry.*
- [LabelComboBox::get](#) (path,...)  
*Procedure to get the [ComboBox](#) value.*
- [LabelComboBox::getlistbox](#) (path,...)  
*Procedure to get the listbox of the [ComboBox](#) widget.*
- [LabelComboBox::getvalue](#) (path,...)  
*Procedure to get the value of the [ComboBox](#).*
- [LabelComboBox::icursor](#) (path,...)  
*Pass through procedure for the [ComboBox](#) icursor function.*
- [LabelComboBox::post](#) (path,...)  
*Pass through procedure for the [ComboBox](#) post function.*
- [LabelComboBox::setvalue](#) (path,...)  
*Pass through procedure for the [ComboBox](#) setvalue function.*
- [LabelComboBox::unpost](#) (path,...)  
*Pass through procedure for the [ComboBox](#) unpost function.*
- [LabelComboBox::\\_path\\_command](#) (path, cmd, larg)  
*Path command for this megawidget.*
- [LabelComboBox::\\_destroy](#) (path)  
*Destructor function.*

## 10.52 Scripts/Common/labelselectcolor.tcl File Reference

### Namespaces

- [LabelSelectColor](#)

*This package provides a BWidget style megawidget for selecting colors, in the same style as a LabelEntry widget.*

### Functions

- [LabelSelectColor::create](#) (path,...)  
*Creation procedure.*
- [LabelSelectColor::ColorPopup](#) (path)  
*Procedure bound to the palette button to select a color.*
- [LabelSelectColor::configure](#) (path,...)  
*Configuration procedure: configure one or more options for this widget.*
- [LabelSelectColor::cget](#) (path, option)  
*Configuration option accessor procedure: access one option directly.*
- [LabelSelectColor::\\_path\\_command](#) (path, cmd, larg)  
*Path command for this megawidget.*
- [LabelSelectColor::\\_destroy](#) (path)  
*Destructor function.*

## 10.53 Scripts/Common/labelspinbox.tcl File Reference

### Namespaces

- [LabelSpinBox](#)

*This is a specialized form of the LabelFrame widget containing a SpinBox Widget.*

### Functions

- [LabelSpinBox::create](#) (path,...)  
*Procedure to create a [LabelSpinBox](#).*
- [LabelSpinBox::configure](#) (path,...)  
*Procedure to configure a [LabelSpinBox](#).*
- [LabelSpinBox::cget](#) (path, option)  
*Procedure to get a configuration option.*
- [LabelSpinBox::setvalue](#) (path,...)  
*Procedure to set the value of the SpinBox.*
- [LabelSpinBox::getvalue](#) (path,...)  
*Procedure to get the value of the SpinBox.*
- [LabelSpinBox::bind](#) (path,...)  
*Procedure to set a binding on the SpinBox entry.*
- [LabelSpinBox::\\_path\\_command](#) (path, cmd, larg)  
*Path command for this megawidget.*
- [LabelSpinBox::\\_destroy](#) (path)  
*Destructor function.*

## 10.54 Scripts/Common/mainwindow.tcl File Reference

### Classes

- class [mainwindow](#)

*A widget that is heavily extended from the BWidget MainFrame widget.*

## 10.55 Scripts/Common/panedw.tcl File Reference

### Namespaces

- [PanedWindow](#)

*A modified version of the BWidget [PanedWindow](#).*

## 10.56 Scripts/Common/ParseXML.tcl File Reference

Parse XML and create a simple DOM tree.

### Classes

- class [SimpleDOMElement](#)

*A simple DOM element coded in Tcl using SNIT.*

- class [ParseXML](#)

*Class to hold an XML tree.*

### 10.56.1 Detailed Description

Parse XML and create a simple DOM tree.

Contains two SNI types, one of which is a simple DOM element, used to hold XML elements.

## 10.57 Scripts/Common/ReadConfiguration.tcl File Reference

### Namespaces

- [ReadConfiguration](#)

*The Read Configuration File code is contained in this namespace.*



## Functions

- [ReadConfiguration::ReadConfiguration](#) (filename, configurationArrayName)  
*This procedure reads in the configuration file named by the filename into the array named by configurationArrayName.*
- [ReadConfiguration::IsEven](#) (i)  
*Checks if its argument is an even number.*
- [ReadConfiguration::WriteConfiguration](#) (filename, configurationArrayName)  
*This procedure writes the configuration contained in configurationArrayName to the file named by the filename.*
- [ReadConfiguration::ConfigurationType](#) (...)  
*This macro defines the body of a snit::type that implements a program's global configuration (or preferences).*

## 10.58 Scripts/Common/snitScrollNotebook.tcl File Reference

### Classes

- class [ScrollTabNotebook](#)  
*Tabbed Notebook with scrollable tabs.*

## 10.59 Scripts/Common/splash.tcl File Reference

### Classes

- class [splash](#)  
*Widget that implements a splash window.*

## 10.60 Scripts/ControlSupport/CmriSupport.tcl File Reference

### Classes

- class [CmriSupport::CmriNode](#)  
*CMR/I node type.*

### Namespaces

- [CmriSupport](#)  
*Cmri Support code.*

## 10.61 Scripts/CTIAcela/CTIAcela.tcl File Reference

### Classes

- class [ctiacela::CTIAcela](#)  
*Main CTIAcela interface class.*

### Namespaces

- [ctiacela](#)  
*CTI Acela Tcl Serial Port Interface.*

### Typedefs

- typedef int [ctiacela::addresstype](#)  
*Module address type.*
- typedef int [ctiacela::ubyte](#)  
*Unsigned byte type.*
- typedef int [ctiacela::speedtype](#)  
*Speed type.*
- typedef int [ctiacela::momtype](#)  
*Momentum control type.*
- typedef int [ctiacela::filterthreshtype](#)  
*Filter threshold type.*

### Variables

- [ctiacela::ctiacela](#)

## 10.62 Scripts/GRSupport/grsupport2.tcl File Reference

### Namespaces

- [GRSupport](#)  
*Code to support the various graphics packages.*

## Functions

- [GRSupport::\\_ROPI2](#) (name1, name2, op)  
*A variable trace to enforce the read-only-ness of PI2.*
- [GRSupport::\\_ROPI](#) (name1, name2, op)  
*A variable trace to enforce the read-only-ness of PI.*
- [GRSupport::DegreesToRadians](#) (degrees)  
*Function to convert from degrees to radians.*
- [GRSupport::RadiansToDegrees](#) (rads)  
*Function to convert from radians to degrees.*
- [GRSupport::VerifyDoubleMethod](#) ()  
*Snit macro defining a validate method for doubles.*
- [GRSupport::VerifyBooleanMethod](#) ()  
*Snit macro defining a validate method for booleans.*
- [GRSupport::VerifyIntegerMethod](#) ()  
*Snit macro defining a validate method integers.*
- [GRSupport::VerifyOrientationHVMMethod](#) ()  
*Snit macro defining a validate method for orientation (horizontal or vertical).*
- [GRSupport::VerifyColorMethod](#) ()  
*Snit macro defining a validate method for colors.*

## Variables

- [GRSupport::PI2](#)  
*Variable containing  $\pi/2$ .*
- [GRSupport::PI](#)  
*Variable containing  $\pi$ .*

## 10.63 Scripts/GRSupport/GRSupportTclGroup.h File Reference

## 10.64 Scripts/GRSupport/Instruments2.tcl File Reference

## Classes

- class [Instruments::DialInstrument](#)  
*Generic dial instrument.*
- class [Instruments::AnalogClock](#)  
*Analog clock instrument.*
- class [Instruments::DigitalInstrument](#)  
*Digital instrument.*
- class [Instruments::DigitalClock](#)  
*Digital clock instrument.*
- class [Instruments::CabSignalLamp](#)  
*Cab signal lamp type.*

## Namespaces

- [Instruments](#)

*Namespace used for instruments code.*

## Functions

- [Instruments::CommonOptions](#) (defaultLabel)

*Snit macro to define common options used by all instruments.*

## 10.65 Scripts/GRSupport/LCARSWidgets2.tcl File Reference

### Namespaces

- [LCARS](#)

*Namespace where the [LCARS](#) code lives.*

## 10.66 Scripts/GRSupport/OvalWidgets2.tcl File Reference

### Classes

- class [OvalWidgets::OvalButton](#)  
*Oval button.*
- class [OvalWidgets::OvalScrollBar](#)  
*Oval ScrollBar.*
- class [OvalWidgets::OvalScale](#)  
*An oval scale widget, much like a standard Tk scale widget.*
- class [OvalWidgets::OvalSlider](#)  
*Oval Slider.*
- class [OvalWidgets::OvalRoundCornerRectangle](#)  
*Oval Round Corner Rectangle.*

### Namespaces

- [OvalWidgets](#)

*These oval shaped widgets are much like the Star Trek NG computer screens.*

## Functions

- [OvalWidgets::XYWH](#) (width, height)  
*Defines the options for position (-x,-y) and size (-width,-height).*
- [OvalWidgets::ColorOptionMethods](#) ()  
*Snit macro to default color option methods.*
- [OvalWidgets::CommonValidateMethods](#) ()  
*Macro to include the common validation methods.*
- [OvalWidgets::ColorFillOption](#) (optspec, default)  
*Method to define a fill color option.*
- [OvalWidgets::ColorOutlineOption](#) (optspec, default)  
*Method to define an outline color option.*
- [OvalWidgets::FontFamily](#) (default)  
*Macro to define the -fontfamily option.*
- [OvalWidgets::SquareEndOptions](#) ()  
*Macro to define the square end options (-rightsquare, -leftsquare).*
- [OvalWidgets::\\_VerifyFont](#) (option, value)  
*Method to validate a font value.*
- [OvalWidgets::\\_ConfigureFont](#) (option, value)  
*Method to configure a font value.*
- [OvalWidgets::\\_ConfigureText](#) (option, value)  
*Method to configure the text of the button.*
- [OvalWidgets::\\_VerifyIntegerOrEmpty](#) (option, value)  
*Method to validate an integer or empty string option.*
- [OvalWidgets::OvalLabel](#) (name, \_canvas,...)  
*Construct some text.*
- [OvalWidgets::~~OvalLabel](#) ()  
*Destructor free up all resources.*
- [OvalWidgets::\\_UnderSplit](#) (beforevar, undervar, aftervar)  
*Method to split label text into before, under, and after segments.*

## Variables

- [OvalWidgets::HBar](#)  
*Holds the horizontal bar bitmap.*
- [OvalWidgets::VBar](#)  
*Holds the vertical bar bitmap.*
- var [OvalWidgets::canvas](#)  
*Canvas the widget is on.*

## 10.67 Scripts/LCC/ConfigDialogs.tcl File Reference

### Classes

- class [lcc::ConfigOptions](#)  
*Display memory config options.*
- class [lcc::ConfigMemory](#)  
*Configure memory.*

## Namespaces

- [lcc](#)

*Namespace that holds the LCC interface code.*

## 10.68 Scripts/LCC/ConfigurationEditor.tcl File Reference

### Classes

- class [lcc::ConfigurationEditor](#)

*Generate OpenLCB Memory Configuration Window.*

## Namespaces

- [lcc](#)

*Namespace that holds the LCC interface code.*

## 10.69 Scripts/LCC/eventDialogs.tcl File Reference

### Classes

- class [lcc::EventLog](#)

*Event received log, with event sender.*

- class [lcc::EventReceived](#)

*Display a received event.*

- class [lcc::SendEvent](#)

*Send Event Dialog – send PCRE message.*

## Namespaces

- [lcc](#)

*Namespace that holds the LCC interface code.*

## 10.70 Scripts/LCC/lcc.tcl File Reference

### Classes

- class [lcc::EventID](#)  
*An event id structure.*
- class [lcc::EventID\\_or\\_null](#)  
*An [EventID](#) or empty string.*
- class [lcc::CANHeader](#)  
*CAN Header type.*
- class [lcc::MTIHeader](#)  
*MTI Header type.*
- class [lcc::MTIDetail](#)  
*MTI Header type, detailed version.*
- class [lcc::CanMessage](#)  
*A CAN Message, containing a 29-bit header and upto 8 bytes of data.*
- class [lcc::GridConnectMessage](#)  
*A Grid Connect formatted CAN message.*
- class [lcc::GridConnectReply](#)  
*A Grid Connect formatted CAN message (reply).*
- class [lcc::nid\\_or\\_null](#)  
*Node ID regexp pattern or the empty string.*
- class [lcc::CanAlias](#)  
*Implements a CAN Alias.*
- class [lcc::CanTransport](#)  
*Logical transport of CAN Messages.*
- class [lcc::OpenLCBMessage](#)  
*OpenLCB Message type.*
- class [lcc::CANGridConnect](#)  
*Base class to connect to a CAN bus using GridConnect formatted message over.*
- class [lcc::CANGridConnectOverUSBSerial](#)  
*Connect to a CAN bus using GridConnect formatted message over a USB Serial port.*
- class [lcc::OpenLCBOverTcp](#)  
*Connect to a OpenLCB over Tcp/Ip.*
- class [lcc::CANGridConnectOverTcp](#)  
*Connect to a CAN bus using GridConnect formatted message over a Tcp/Ip connection.*
- class [lcc::CANGridConnectOverCANSocket](#)  
*Connect to a CAN bus using GridConnect formatted message over a CAN Socket connection.*
- class [lcc::OpenLCBProtocols](#)  
*Supported LCC Protocol name type.*
- class [lcc::OpenLCBNode](#)  
*Connect to a OpenLCB interface.*

### Namespaces

- [lcc](#)  
*Namespace that holds the LCC interface code.*

## Typedefs

- typedef int [lcc::twobits](#)  
*A 2 bit integer.*
- typedef int [lcc::threebits](#)  
*A 3 bit integer.*
- typedef int [lcc::fivebits](#)  
*A 5 bit integer.*
- typedef int [lcc::sixbits](#)  
*A 6 bit integer.*
- typedef int [lcc::length](#)  
*An integer from 1 to 64.*
- typedef int [lcc::byte](#)  
*An 8-bit unsigned byte.*
- typedef int [lcc::twelvebits](#)  
*A 12 bit integer.*
- typedef int [lcc::fifteenbits](#)  
*A 15 bit integer.*
- typedef int [lcc::sixteenbits](#)  
*A 16 bit integer.*
- typedef int [lcc::headerword](#)  
*A 29 bit integer.*
- typedef int [lcc::uint32](#)  
*A 32 bit unsigned integer.*
- typedef listtype [lcc::eightbytes](#)  
*A list of bytes, from 0 to 8 elements.*
- typedef listtype [lcc::bytelist72](#)  
*A list of bytes, from 0 to 72 elements.*
- typedef listtype [lcc::bytelist](#)  
*A list of bytes, unbounded.*
- typedef listtype [lcc::databuf](#)  
*A list of bytes, from 1 to 64 elements.*
- typedef char \* [lcc::nid](#)  
*Node ID regexp pattern.*

## Enumerations

- enum [lcc::datagramcontent](#) {  
    [lcc::complete](#), [lcc::first](#), [lcc::middle](#), [lcc::last](#),  
    [lcc::stream](#) }  
*Datagram and stream types.*
- enum [lcc::eventvalidity](#) { [lcc::valid](#), [lcc::invalid](#), [lcc::unknown](#) }  
*Event validity.*



## Functions

- [lcc::AbstractMessage](#) ()  
*Define common variables and accessor methods.*
- [lcc::AbstractMRMessage](#) ()  
*Macro to create common methods and variables for an AbstractMRMessage*

## 10.71 Scripts/LinuxGpio/LinuxGpio.tcl File Reference

### Classes

- class [linuxgpio::LinuxGpio](#)  
*Base generic GPIO interface class.*
- class [linuxgpio::GpioOutputSafeLow](#)  
*Output pin, initialized to low.*
- class [linuxgpio::GpioOutputSafeHigh](#)  
*Output pin, initialized to high.*
- class [linuxgpio::GpioOutputSafeLowInverted](#)  
*Output pin, initialized to low, with inverted logic.*
- class [linuxgpio::GpioOutputSafeHighInvert](#)  
*Output pin, initialized to high, inverted.*
- class [linuxgpio::GpioInputActiveHigh](#)  
*Input pin, active high (high is true).*
- class [linuxgpio::GpioInputActiveLow](#)  
*Input pin, active low (low is true).*

### Namespaces

- [linuxgpio](#)  
*Linux GPIO Interface.*

### Typedefs

- typedef int [linuxgpio::pinnotype](#)  
*Pin number type, a positive integer.*

### Enumerations

- enum [linuxgpio::pindirection](#) { [linuxgpio::in](#), [linuxgpio::out](#), [linuxgpio::high](#), [linuxgpio::low](#) }  
*Pin direction and initial type code.*

## 10.72 Scripts/NCE/nce.tcl File Reference

### Classes

- class [nce::NCE](#)  
*Main [NCE](#) Cab Bus interface class.*

### Namespaces

- [nce](#)  
*Namespace that holds the [NCE](#) interface code.*

### Typedefs

- typedef int [nce::LocoAddress](#)  
*Locomotive address type.*
- typedef int [nce::ConsistAddress](#)  
*Consist address type.*
- typedef int [nce::AccessoryNumber](#)  
*Accessory address type.*
- typedef int [nce::MacroNumber](#)  
*[NCE](#) Macro number.*
- typedef int [nce::CabNumber](#)  
*Cab number type.*
- typedef int [nce::Hours](#)  
*Hours type.*
- typedef int [nce::Minutes](#)  
*Minutes type.*
- typedef int [nce::ScaleClockRatio](#)  
*Scale clock ratio range.*
- typedef int [nce::EchoMode](#)  
*This is really should be an enumeration, but works as a limited range integer.*
- typedef int [nce::Speed28](#)  
*28 speed step speeds.*
- typedef int [nce::Speed128](#)  
*128 speed step speeds.*
- typedef int [nce::CSAddress](#)  
*CSAddress type.*
- typedef int [nce::UByte](#)  
*Unsigned byte type (data byte).*
- typedef listtype [nce::RAMData](#)  
*Datalist for RAM data, 1 to 16 unsigned bytes.*
- typedef char \* [nce::LCDMessage16](#)  
*Data for 16 character LCD lines.*
- typedef char \* [nce::LCDMessage8](#)

*Data for 8 character LCD lines.*

- typedef listtype [nce::RawPacket](#)

*Raw packets for writing raw packets to the temp queue.*

- typedef listtype [nce::RawTrackPacket](#)

*Raw packets for writing raw packets to the track queue.*

- typedef listtype [nce::RAMData8](#)

*Datalist for RAM data 8 unsigned bytes.*

- typedef int [nce::MomentumLevel](#)

*Momentum level.*

- typedef int [nce::AspectBits](#)

*Aspect bit mask.*

## Enumerations

- enum [nce::SpeedMode](#) { [nce::S14](#), [nce::S28](#), [nce::S128](#) }
- enum [nce::Direction](#) { [nce::Forward](#), [nce::Reverse](#) }

## Functions

- [nce::ErrorMessage](#) (code)  
*Return the error message, given the error code.*

## 10.73 Scripts/RailDriverSupport/raildriver\_client.tcl File Reference

### Classes

- class [raildriver::RaildriverClient](#)  
*Raildriver Client class – connects to the Raildriver daemon.*

### Namespaces

- [raildriver](#)  
*Namespace that holds the Raildriver Client class code.*

### Typedefs

- typedef listtype [raildriver::eventlist](#)  
*List of event codes.*

## Enumerations

- enum [raildriver::RaildriverEvents](#) {  
[raildriver::REVERSER](#), [raildriver::THROTTLE](#), [raildriver::AUTOBRAKE](#), [raildriver::INDEPENDBRK](#),  
[raildriver::BAILOFF](#), [raildriver::WIPER](#), [raildriver::HEADLIGHT](#), [raildriver::DIGITAL1](#),  
[raildriver::DIGITAL2](#), [raildriver::DIGITAL3](#), [raildriver::DIGITAL4](#), [raildriver::DIGITAL5](#),  
[raildriver::DIGITAL6](#) }

*These are the event codes for the Rail Driver's report message.*

## 10.74 Scripts/Satellite/Satellite.tcl File Reference

### Classes

- class [Satellite](#)  
[Satellite](#) class.

## 10.75 Scripts/XPressNet/xpressnet.tcl File Reference

### Classes

- class [xpressnet::CommandStationResponse](#)  
*General response class.*
- class [xpressnet::ServiceModeResponse](#)  
*Service mode response.*
- class [xpressnet::SoftwareVersion](#)  
*Software version.*
- class [xpressnet::CommandStationStatus](#)  
*Command station status.*
- class [xpressnet::AccessoryDecoderInformation](#)  
*Accessory decoder information.*
- class [xpressnet::LocomotiveInformation](#)  
*Locomotive information.*
- class [xpressnet::FunctionStatus](#)  
*Function status.*
- class [xpressnet::LocomotiveAddress](#)  
*Locomotive address.*
- class [xpressnet::DoubleHeaderInformation](#)  
*Double header information.*
- class [xpressnet::DoubleHeaderMuError](#)  
*Double header or MU error.*
- class [xpressnet::LI100Message](#)  
*LI100 messages.*
- class [xpressnet::LI100VersionNumbers](#)  
*LI100 Version Numbers.*
- class [xpressnet::LI101XPressNetAddress](#)  
*LI101 XPress Net Address.*
- class [xpressnet::XPressNet](#)  
*Main XPressNet interface class.*
- class [xpressnet::XpressNetEvent](#)  
[XPressNet](#) Event class.

## Namespaces

- [xpressnet](#)

*Namespace that holds the [XPressNet](#) interface code.*

## Typedefs

- typedef int [xpressnet::nibble](#)  
*A 4 bit unsigned integer.*
- typedef int [xpressnet::ubyte](#)  
*An 8 bit unsigned integer.*
- typedef int [xpressnet::DecoderLongAddress](#)  
*Decoder address, an unsigned 14 bit integer.*
- typedef int [xpressnet::ElementAddress](#)  
*A 2 bit unsigned integer.*
- typedef int [xpressnet::S\\_14](#)  
*14 Speed steps.*
- typedef int [xpressnet::S\\_27](#)  
*27 Speed steps.*
- typedef int [xpressnet::S\\_28](#)  
*28 Speed steps.*
- typedef int [xpressnet::S\\_128](#)  
*128 Speed steps.*
- typedef int [xpressnet::u10](#)  
*An unsigned 10 bit integer.*
- typedef int [xpressnet::u3](#)  
*An unsigned 3 bit integer.*
- typedef int [xpressnet::u7](#)  
*An unsigned 7 bit integer.*
- typedef int [xpressnet::ConsistAddress](#)  
*Multi-unit Address.*

## Enumerations

- enum [xpressnet::TypeCode](#) {  
[xpressnet::NO\\_RESPONSE\\_AVAILABLE](#), [xpressnet::NORMAL\\_OPERATION\\_RESUMED](#), [xpressnet::TRAC←](#)  
[K\\_POWER\\_OFF](#), [xpressnet::EMERGENCY\\_STOP](#),  
[xpressnet::SERVICE\\_MODE\\_ENTRY](#), [xpressnet::PROGRAMMING\\_INFO\\_SHORT\\_CIRCUIT](#), [xpressnet::PR←](#)  
[OGRAMMING\\_INFO\\_DATA\\_BYTE\\_NOT\\_FOUND](#), [xpressnet::PROGRAMMING\\_INFO\\_COMMAND\\_STATIO←](#)  
[N\\_BUSY](#),  
[xpressnet::PROGRAMMING\\_INFO\\_COMMAND\\_STATION\\_READY](#), [xpressnet::SERVICE\\_MODE\\_RESPON←](#)  
[SE](#), [xpressnet::SOFTWARE\\_VERSION](#), [xpressnet::COMMAND\\_STATION\\_STATUS](#),  
[xpressnet::TRANSFER\\_ERRORS](#), [xpressnet::COMMAND\\_STATION\\_BUSY](#), [xpressnet::INSTRUCTION\\_NO←](#)  
[T\\_SUPPORTED](#), [xpressnet::ACCESSORY\\_DECODER\\_INFORMATION](#),  
[xpressnet::LOCOMOTIVE\\_INFORMATION](#), [xpressnet::FUNCTION\\_STATUS](#), [xpressnet::LOCOMOTIVE\\_AD←](#)  
[DRESS](#), [xpressnet::DOUBLE\\_HEADER\\_INFORMATION](#),  
[xpressnet::DOUBLE\\_HEADER\\_MU\\_ERROR](#), [xpressnet::LI100\\_MESSAGE](#), [xpressnet::LI100\\_VERSION](#),  
[xpressnet::LI101\\_XPRESSNET\\_ADDRESS](#) }

*Response types.*

- enum `xpressnet::PowerUpMode` { `xpressnet::Manual`, `xpressnet::Automatic` }

*Power up modes.*

- enum `xpressnet::NibbleCode` { `xpressnet::Lower`, `xpressnet::Upper` }

*Accessory nibble code.*

- enum `xpressnet::SpeedStepModeCode` { `xpressnet::S14`, `xpressnet::S27`, `xpressnet::S28`, `xpressnet::S128` }

*Speed step mode code.*

- enum `xpressnet::DirectionCode` { `xpressnet::Forward`, `xpressnet::Reverse` }

*Direction flag.*

- enum `xpressnet::ErrorTypeCode` {  
`xpressnet::NotOperatedOr0`, `xpressnet::UsedByAnotherDevice`, `xpressnet::UsedInANotherDHMU`, `xpressnet::↵  
SpeedNotZero`,  
`xpressnet::NotMU`, `xpressnet::NotMUBaseAddress`, `xpressnet::CantDelete`, `xpressnet::StackFull` }

*Error type code.*

- enum `xpressnet::MessageTypeCode` {  
`xpressnet::ErrorBetweenLI100AndPC`, `xpressnet::ErrorBetweenLI100AndCommandStation`, `xpressnet::↵  
UnknownCommunicationsError`, `xpressnet::Success`,  
`xpressnet::NoTimeslot`, `xpressnet::BufferOverflow`, `xpressnet::Other` }

*Message type code.*

# Index

- - gettext, [98](#)
- \_ADDRESS
  - xpressnet::XPressNet, [1256](#)
- \_AddTLever
  - CTCPanel::Toggle, [1156](#)
- \_BaseRect
  - OvalWidgets::OvalScale, [716](#)
  - OvalWidgets::OvalScrollBar, [725](#)
- \_CancelOpenTransport
  - lcc::CANGridConnectOverCANSocket, [217](#)
  - lcc::CANGridConnectOverTcp, [222](#)
  - lcc::CANGridConnectOverUSBSerial, [226](#)
  - lcc::OpenLCBOverTcp, [702](#)
- \_CancelSelectTransport
  - lcc::OpenLCBNode, [690](#)
- \_CheckForResponse\_0x00
  - xpressnet::XPressNet, [1256](#)
- \_CheckForResponse\_0x40
  - xpressnet::XPressNet, [1257](#)
- \_CheckForResponse\_0x60
  - xpressnet::XPressNet, [1257](#)
- \_CheckForResponse\_0x80
  - xpressnet::XPressNet, [1257](#)
- \_CheckForResponse\_0xa0
  - xpressnet::XPressNet, [1257](#)
- \_CheckForResponse\_0xc0
  - xpressnet::XPressNet, [1257](#)
- \_CheckForResponse\_0xe0
  - xpressnet::XPressNet, [1258](#)
- \_Close
  - HTMLHelp::HTMLHelp, [489](#)
  - lcc::ConfigMemory, [305](#)
  - lcc::ConfigOptions, [313](#)
  - lcc::EventReceived, [442](#)
  - lcc::SendEvent, [887](#)
- \_Command
  - OvalWidgets::OvalScrollBar, [726](#)
- \_Configure
  - ScrollTabNotebook, [866](#)
- \_ConfigureL
  - OvalWidgets::OvalSlider, [721](#)
- \_ConfigureFillColor
  - Instruments::CabSignalLamp, [198](#)
- \_ConfigureFont
  - OvalWidgets, [136](#)
- \_ConfigureOutlineColor
  - Instruments::CabSignalLamp, [198](#)
- \_ConfigureSize
  - Instruments::CabSignalLamp, [198](#)
- \_ConfigureText
  - OvalWidgets, [136](#)
  - OvalWidgets::OvalButton, [711](#)
  - OvalWidgets::OvalScale, [716](#)
  - OvalWidgets::OvalSlider, [721](#)
- \_ConfigureWL
  - OvalWidgets::OvalScale, [716](#)
  - OvalWidgets::OvalScrollBar, [726](#)
- \_ConfigureXY
  - Instruments::CabSignalLamp, [199](#)
  - OvalWidgets::OvalScale, [717](#)
  - OvalWidgets::OvalScrollBar, [726](#)
- \_CtcMainHScroll2
  - CTCPanel::CTCPanel, [371](#)
- \_CtcMainSyncX
  - CTCPanel::CTCPanel, [371](#)
- \_Dump
  - lcc::ConfigMemory, [306](#)
- \_MoveTLever
  - CTCPanel::Toggle, [1157](#)
- \_MoveThumb
  - OvalWidgets::OvalScale, [717](#)
  - OvalWidgets::OvalSlider, [721](#)
  - OvalWidgets::OvalScrollBar, [727](#)
- \_OpenTransport
  - lcc::CANGridConnectOverCANSocket, [217](#)
  - lcc::CANGridConnectOverTcp, [222](#)
  - lcc::CANGridConnectOverUSBSerial, [226](#)
  - lcc::OpenLCBOverTcp, [702](#)
- \_PI
  - CTCPanel::CurvedBlock, [398](#)
- \_PlatePolygon
  - CTCPanel::SIGPlate, [901](#)
  - CTCPanel::SWPlate, [1023](#)
- \_PosInteger
  - CTCPanel::CTCPanel, [372](#)
- \_RAM\_check\_error
  - xpressnet::CommandStationStatus, [299](#)
- \_ROPI
  - GRSupport, [99](#)

- [\\_ROPI2](#)
  - [GRSupport, 100](#)
- [\\_RTRoffset](#)
  - [lcc::GridConnectReply, 478](#)
- [\\_RadiansToDegrees](#)
  - [CTCPanel::CurvedBlock, 396](#)
- [\\_Read](#)
  - [lcc::ConfigMemory, 307](#)
- [\\_Restore](#)
  - [lcc::ConfigMemory, 308](#)
- [\\_SBackward](#)
  - [HTMLHelp::HTMLHelp, 489](#)
- [\\_SForward](#)
  - [HTMLHelp::HTMLHelp, 489](#)
- [\\_SchematicDrawOval](#)
  - [CTCPanel::Signal, 894](#)
- [\\_SchematicDrawThinLine](#)
  - [CTCPanel::Signal, 895](#)
- [\\_SelectTransport](#)
  - [lcc::OpenLCBNode, 691](#)
- [\\_Send](#)
  - [lcc::SendEvent, 887](#)
- [\\_StubYard\\_Poly](#)
  - [CTCPanel::StubYard, 1000](#)
- [\\_ThroughYard\\_Poly](#)
  - [CTCPanel::ThroughYard, 1123](#)
- [\\_TypeCodes](#)
  - [CmriSupport::CmriNode, 290](#)
- [\\_UnderSplit](#)
  - [OvalWidgets, 137](#)
- [\\_ValidateAddress](#)
  - [CmriSupport::CmriNode, 285](#)
- [\\_ValidateByte](#)
  - [CmriSupport::CmriNode, 285](#)
- [\\_ValidateListOfBytes](#)
  - [CmriSupport::CmriNode, 287](#)
- [\\_ValidateSixElementListOfBytes](#)
  - [CmriSupport::CmriNode, 287](#)
- [\\_ValidateType](#)
  - [CmriSupport::CmriNode, 287](#)
- [\\_ValidateWord](#)
  - [CmriSupport::CmriNode, 288](#)
- [\\_VerifyBitmap](#)
  - [OvalWidgets::OvalSlider, 722](#)
- [\\_VerifyCrossingType](#)
  - [CTCPanel::Crossing, 351](#)
- [\\_VerifyFont](#)
  - [OvalWidgets, 137](#)
- [\\_VerifyHeads](#)
  - [CTCPanel::Signal, 895](#)
- [\\_VerifyIntegerOrEmpty](#)
  - [OvalWidgets, 137](#)
- [\\_VerifyOrientationHV](#)
  - [CTCPanel::Toggle, 1158](#)
- [\\_WidgetMap](#)
  - [HTMLHelp::HTMLHelp, 509](#)
- [\\_Write](#)
  - [lcc::ConfigMemory, 308](#)
- [\\_accessory\\_type](#)
  - [xpressnet::AccessoryDecoderInformation, 165](#)
- [\\_addnewloco](#)
  - [CabWidgets::SelectLocomotive, 885](#)
- [\\_address](#)
  - [xpressnet::AccessoryDecoderInformation, 165](#)
  - [xpressnet::DoubleHeaderInformation, 419](#)
  - [xpressnet::LI101XPressNetAddress, 569](#)
  - [xpressnet::LocomotiveAddress, 575](#)
  - [xpressnet::LocomotiveInformation, 584](#)
- [\\_address2](#)
  - [xpressnet::DoubleHeaderInformation, 419](#)
  - [xpressnet::LocomotiveInformation, 584](#)
- [\\_addressType](#)
  - [xpressnet::LocomotiveAddress, 575](#)
- [\\_appendXORByte](#)
  - [xpressnet::XPressNet, 1256](#)
- [\\_available](#)
  - [xpressnet::DoubleHeaderInformation, 419](#)
  - [xpressnet::LocomotiveInformation, 584](#)
- [\\_buildSelectTransportConstructorDialog](#)
  - [lcc::OpenLCBNode, 690](#)
- [\\_cgetdata](#)
  - [lcc::OpenLCBMessage, 685](#)
- [\\_ch\\_oldfocus](#)
  - [CTCPanel::CTCPanel, 388](#)
- [\\_ch\\_oldgrab](#)
  - [CTCPanel::CTCPanel, 388](#)
- [\\_characterdata](#)
  - [ParseXML, 756](#)
- [\\_children](#)
  - [SimpleDOMElement, 909](#)
- [\\_clear](#)
  - [lcc::EventLog, 439](#)
- [\\_clientRow](#)
  - [ScrollTabNotebook, 874](#)
- [\\_close](#)
  - [lcc::ConfigurationEditor, 319](#)
  - [lcc::EventLog, 439](#)
- [\\_command\\_station\\_type](#)
  - [xpressnet::SoftwareVersion, 923](#)
- [\\_completed](#)
  - [xpressnet::AccessoryDecoderInformation, 165](#)
- [\\_compute\\_height](#)
  - [ScrollTabNotebook, 866](#)
- [\\_compute\\_width](#)
  - [ScrollTabNotebook, 866](#)
- [\\_configureCenterLabel](#)
  - [CTCPanel::Toggle, 1157](#)
- [\\_configureColor](#)



- CTCPanel::CTCLabel, [363](#)
- CTCPanel::Lamp, [548](#)
- CTCPanel::SchLabel, [856](#)
- \_configureLabel
  - CTCPanel::CTCLabel, [364](#)
  - CTCPanel::Crossing, [351](#)
  - CTCPanel::Crossover, [355](#)
  - CTCPanel::CurvedBlock, [396](#)
  - CTCPanel::DoubleSlip, [426](#)
  - CTCPanel::EndBumper, [432](#)
  - CTCPanel::HiddenBlock, [481](#)
  - CTCPanel::Lamp, [548](#)
  - CTCPanel::PushButton, [812](#)
  - CTCPanel::SIGPlate, [900](#)
  - CTCPanel::SWPlate, [1021](#)
  - CTCPanel::SchLabel, [856](#)
  - CTCPanel::ScissorCrossover, [860](#)
  - CTCPanel::Signal, [894](#)
  - CTCPanel::SingleSlip, [911](#)
  - CTCPanel::StraightBlock, [995](#)
  - CTCPanel::StubYard, [999](#)
  - CTCPanel::Switch, [1003](#)
  - CTCPanel::ThreeWaySW, [1118](#)
  - CTCPanel::ThroughYard, [1122](#)
- \_configureLeftLabel
  - CTCPanel::Toggle, [1157](#)
- \_configureRightLabel
  - CTCPanel::Toggle, [1157](#)
- \_configuredata
  - lcc::OpenLCBMessage, [686](#)
- \_copyCM
  - lcc::GridConnectMessage, [468](#)
- \_copyGCM
  - lcc::GridConnectReply, [474](#)
- \_crosshairEnd
  - CTCPanel::CTCPanel, [370](#)
- \_crosshairMove
  - CTCPanel::CTCPanel, [370](#)
- \_crosshairStart
  - CTCPanel::CTCPanel, [371](#)
- \_cv
  - xpressnet::ServiceModeResponse, [890](#)
- \_data
  - SimpleDOMEElement, [909](#)
  - xpressnet::ServiceModeResponse, [890](#)
- \_datagramhandler
  - lcc::ConfigMemory, [305](#)
  - lcc::ConfigurationEditor, [319](#)
- \_datagramrejecterror
  - lcc::ConfigMemory, [309](#)
  - lcc::ConfigurationEditor, [335](#)
- \_destroy
  - FileEntry, [95](#)
  - LabelComboBox, [106](#)
  - LabelSelectColor, [111](#)
  - LabelSpinBox, [116](#)
- \_direction
  - CabWidgets::LocomotiveDirection, [578](#)
  - xpressnet::DoubleHeaderInformation, [419](#)
  - xpressnet::LocomotiveInformation, [584](#)
- \_down
  - CabWidgets::LocomotiveSpeed, [591](#)
- \_down1
  - CabWidgets::LocomotiveSpeed, [590](#)
- \_down10
  - CabWidgets::LocomotiveSpeed, [590](#), [592](#)
- \_draw\_arrows
  - ScrollTabNotebook, [867](#)
- \_draw\_page
  - ScrollTabNotebook, [867](#)
- \_dumpAsHex
  - lcc::ConfigMemory, [306](#)
- \_dumpAsText
  - lcc::ConfigMemory, [306](#)
- \_elementend
  - ParseXML, [756](#)
- \_elementstart
  - ParseXML, [756](#)
- \_emergency\_off
  - xpressnet::CommandStationStatus, [298](#)
- \_emergency\_stop
  - xpressnet::CommandStationStatus, [298](#)
- \_error
  - xpressnet::DoubleHeaderMuError, [423](#)
- \_eventID
  - lcc::EventID, [437](#)
- \_eventhandler
  - xpressnet::XpressNetEvent, [1274](#)
- \_eventidComboRead
  - lcc::ConfigurationEditor, [320](#)
- \_eventidComboWrite
  - lcc::ConfigurationEditor, [320](#)
- \_eventidEntryRead
  - lcc::ConfigurationEditor, [320](#)
- \_eventidEntryWrite
  - lcc::ConfigurationEditor, [321](#)
- \_eventidnumber
  - lcc::ConfigurationEditor, [335](#)
- \_explodechars
  - ncc::NCE, [649](#)
- \_flags0
  - lcc::CANGridConnect, [206](#)
- \_formatrlist
  - SimpleDOMEElement, [903](#)
- \_function0
  - xpressnet::DoubleHeaderInformation, [419](#)
  - xpressnet::LocomotiveInformation, [585](#)
- \_function1

- xpressnet::DoubleHeaderInformation, [420](#)
- xpressnet::LocomotiveInformation, [585](#)
- \_function10
  - xpressnet::DoubleHeaderInformation, [420](#)
  - xpressnet::LocomotiveInformation, [585](#)
- \_function11
  - xpressnet::DoubleHeaderInformation, [420](#)
  - xpressnet::LocomotiveInformation, [585](#)
- \_function12
  - xpressnet::DoubleHeaderInformation, [420](#)
  - xpressnet::LocomotiveInformation, [585](#)
- \_function2
  - xpressnet::DoubleHeaderInformation, [420](#)
  - xpressnet::LocomotiveInformation, [585](#)
- \_function3
  - xpressnet::DoubleHeaderInformation, [420](#)
  - xpressnet::LocomotiveInformation, [586](#)
- \_function4
  - xpressnet::DoubleHeaderInformation, [421](#)
  - xpressnet::LocomotiveInformation, [586](#)
- \_function5
  - xpressnet::DoubleHeaderInformation, [421](#)
  - xpressnet::LocomotiveInformation, [586](#)
- \_function6
  - xpressnet::DoubleHeaderInformation, [421](#)
  - xpressnet::LocomotiveInformation, [586](#)
- \_function7
  - xpressnet::DoubleHeaderInformation, [421](#)
  - xpressnet::LocomotiveInformation, [586](#)
- \_function8
  - xpressnet::DoubleHeaderInformation, [421](#)
  - xpressnet::LocomotiveInformation, [586](#)
- \_function9
  - xpressnet::DoubleHeaderInformation, [421](#)
  - xpressnet::LocomotiveInformation, [587](#)
- \_getAddressRange
  - lcc::ConfigMemory, [307](#)
- \_getEventID
  - lcc::EventID, [435](#)
- \_get\_extended
  - lcc::GridConnectMessage, [468](#)
  - lcc::GridConnectReply, [474](#)
- \_get\_rtr
  - lcc::GridConnectMessage, [469](#)
  - lcc::GridConnectReply, [475](#)
- \_get\_x\_page
  - ScrollTabNotebook, [867](#)
- \_groupnumber
  - lcc::ConfigurationEditor, [335](#)
- \_handleSRQ
  - CTIAcela, [46](#)
- \_hardware\_version
  - xpressnet::LI100VersionNumbers, [568](#)
- \_header
  - lcc::CanMessage, [239](#)
- \_highlight
  - ScrollTabNotebook, [868](#)
- \_hpage
  - ScrollTabNotebook, [874](#)
- \_intComboRead
  - lcc::ConfigurationEditor, [321](#)
- \_intComboWrite
  - lcc::ConfigurationEditor, [322](#)
- \_intSpinRead
  - lcc::ConfigurationEditor, [322](#)
- \_intSpinWrite
  - lcc::ConfigurationEditor, [323](#)
- \_intnumber
  - lcc::ConfigurationEditor, [335](#)
- \_ioComplete
  - lcc::ConfigMemory, [309](#)
  - lcc::ConfigurationEditor, [336](#)
- \_lastSet
  - OvalWidgets::OvalScrollBar, [729](#)
- \_left
  - CabWidgets::LocomotiveDirection, [578](#)
  - ScrollTabNotebook, [875](#)
- \_m
  - gettext, [98](#)
- \_major
  - xpressnet::SoftwareVersion, [923](#)
- \_menu
  - lcc::ConfigurationEditor, [336](#)
- \_messageHandler
  - lcc::OpenLCBNode, [691](#)
- \_messageReader
  - lcc::CANGridConnect, [206](#)
- \_message\_type
  - xpressnet::LI100Message, [566](#)
- \_messagehandler
  - lcc::ConfigMemory, [307](#)
- \_minor
  - xpressnet::SoftwareVersion, [923](#)
- \_mkbuttons
  - lcc::ConfigurationEditor, [336](#)
- \_mtraddress
  - xpressnet::LocomotiveInformation, [587](#)
- \_mx
  - gettext, [98](#)
- \_nibble
  - xpressnet::AccessoryDecoderInformation, [166](#)
- \_numberOfFeedbackElements
  - xpressnet::AccessoryDecoderInformation, [166](#)
- \_openFile
  - FileEntry, [95](#)
- \_paddingtype
  - ScrollTabNotebook, [875](#)
- \_path\_command

- FileEntry, 95
- LabelComboBox, 106
- LabelSelectColor, 112
- LabelSpinBox, 116
- \_peelId
  - lcc::CanAlias, 201
- \_poller
  - raildriver::RaildriverClient, 815
- \_poweringup
  - xpressnet::CommandStationStatus, 299
- \_printexport
  - lcc::ConfigurationEditor, 323
- \_printexport\_csv
  - lcc::ConfigurationEditor, 323
- \_printexport\_csv\_frame
  - lcc::ConfigurationEditor, 324
- \_printexport\_csv\_frameAcross
  - lcc::ConfigurationEditor, 324
- \_printexport\_csv\_framesAcross
  - lcc::ConfigurationEditor, 325
- \_printexport\_csv\_vframe
  - lcc::ConfigurationEditor, 325
- \_printexport\_csv\_vframeAcross
  - lcc::ConfigurationEditor, 325
- \_printexport\_pdf
  - lcc::ConfigurationEditor, 326
- \_printexport\_pdf\_frame
  - lcc::ConfigurationEditor, 326
- \_printexport\_pdf\_newpage
  - lcc::ConfigurationEditor, 327
- \_printexport\_pdf\_vframe
  - lcc::ConfigurationEditor, 327
- \_printexport\_txt
  - lcc::ConfigurationEditor, 328
- \_printexport\_txt\_frame
  - lcc::ConfigurationEditor, 328
- \_printexport\_txt\_vframe
  - lcc::ConfigurationEditor, 329
- \_printexport\_xml
  - lcc::ConfigurationEditor, 329
- \_printexport\_xml\_frame
  - lcc::ConfigurationEditor, 329
- \_printexport\_xml\_vframe
  - lcc::ConfigurationEditor, 331
- \_processXMLnode
  - lcc::ConfigurationEditor, 331
- \_quoteXML
  - SimpleDOMElement, 904
- \_radiustype
  - ScrollTabNotebook, 875
- \_readall
  - lcc::ConfigurationEditor, 332, 336
- \_readbyte
  - CTIAcela, 46
- cmri::CMri, 278
- nce::NCE, 649
- xpressnet::XPressNet, 1258
- \_readevent
  - CTIAcela, 47
  - cmri::CMri, 279
  - nce::NCE, 650
  - raildriver::RaildriverClient, 815
  - xpressnet::XPressNet, 1258
- \_readmemory
  - lcc::ConfigMemory, 307
  - lcc::ConfigurationEditor, 332
- \_readresponse
  - nce::NCE, 650
- \_redraw
  - ScrollTabNotebook, 868
- \_reserveMyAlias
  - lcc::CANGridConnect, 206
- \_resize
  - ScrollTabNotebook, 868
- \_right
  - CabWidgets::LocomotiveDirection, 579
  - ScrollTabNotebook, 875
- \_script
  - xpressnet::XpressNetEvent, 1274
- \_segmentnumber
  - lcc::ConfigurationEditor, 336
- \_select
  - ScrollTabNotebook, 868
- \_sendDatagram
  - lcc::CANGridConnect, 207
- \_sendMessageAndReturnResponse
  - nce::NCE, 650
- \_sendmessage
  - lcc::CANGridConnect, 207
- \_sendtheevent
  - lcc::EventLog, 439
- \_service\_mode
  - xpressnet::CommandStationStatus, 299
  - xpressnet::ServiceModeResponse, 890
- \_setEventID
  - lcc::EventID, 436
- \_set\_extended
  - lcc::GridConnectMessage, 469
- \_set\_rtr
  - lcc::GridConnectMessage, 469
- \_setdirection
  - CabWidgets::LocomotiveDirection, 577
- \_setspeed
  - CabWidgets::LocomotiveSpeed, 590
- \_software\_version
  - xpressnet::LI100VersionNumbers, 568
- \_spaces
  - lcc::ConfigMemory, 309

- `_speed`
  - `CabWidgets::LocomotiveSpeed`, 592
  - `xpressnet::DoubleHeaderInformation`, 422
  - `xpressnet::LocomotiveInformation`, 587
- `_speedstep`
  - `xpressnet::DoubleHeaderInformation`, 422
  - `xpressnet::LocomotiveInformation`, 587
- `_square`
  - `CTCPanel::CurvedBlock`, 397
- `_start_mode`
  - `xpressnet::CommandStationStatus`, 299
- `_status0`
  - `xpressnet::FunctionStatus`, 450
- `_status1`
  - `xpressnet::FunctionStatus`, 450
- `_status10`
  - `xpressnet::FunctionStatus`, 450
- `_status11`
  - `xpressnet::FunctionStatus`, 450
- `_status12`
  - `xpressnet::FunctionStatus`, 450
- `_status2`
  - `xpressnet::FunctionStatus`, 451
- `_status3`
  - `xpressnet::FunctionStatus`, 451
- `_status4`
  - `xpressnet::FunctionStatus`, 451
- `_status5`
  - `xpressnet::FunctionStatus`, 451
- `_status6`
  - `xpressnet::FunctionStatus`, 451
- `_status7`
  - `xpressnet::FunctionStatus`, 451
- `_status8`
  - `xpressnet::FunctionStatus`, 452
- `_status9`
  - `xpressnet::FunctionStatus`, 452
- `_stop`
  - `CabWidgets::LocomotiveSpeed`, 590, 592
- `_stringComboRead`
  - `lcc::ConfigurationEditor`, 332
- `_stringComboWrite`
  - `lcc::ConfigurationEditor`, 333
- `_stringEntryRead`
  - `lcc::ConfigurationEditor`, 333
- `_stringEntryWrite`
  - `lcc::ConfigurationEditor`, 334
- `_stringnumber`
  - `lcc::ConfigurationEditor`, 336
- `_t1`
  - `xpressnet::AccessoryDecoderInformation`, 166
- `_t2`
  - `xpressnet::AccessoryDecoderInformation`, 166
- `_tabrow`
  - `ScrollTabNotebook`, 875
- `_tabsides`
  - `ScrollTabNotebook`, 875
- `_test_page`
  - `ScrollTabNotebook`, 869
- `_textid`
  - `ScrollTabNotebook`, 876
- `_themeChanged`
  - `ScrollTabNotebook`, 869
- `_themeChanged_`
  - `ScrollTabNotebook`, 869
- `_time_stamp`
  - `xpressnet::CommandStationResponse`, 295
- `_timedout`
  - `lcc::CANGridConnect`, 207
- `_timeout`
  - `CTIAcela`, 61
  - `cmri::CMri`, 281
  - `lcc::CANGridConnect`, 212
  - `lcc::OpenLCBOverTcp`, 705
  - `nce::NCE`, 672
  - `xpressnet::XPressNet`, 1272
- `_timeoutFlag`
  - `lcc::CANGridConnect`, 213
- `_timeoutevent`
  - `nce::NCE`, 650
  - `xpressnet::XPressNet`, 1258
- `_transmit`
  - `CTIAcela`, 47
  - `cmri::CMri`, 279
  - `nce::NCE`, 650
  - `xpressnet::XPressNet`, 1259
- `_transportConstructors`
  - `lcc::OpenLCBNode`, 699
- `_transportlayerconf`
  - `lcc::CanTransport`, 240
- `_trimList`
  - `CabWidgets::SelectLocomotive`, 885
- `_up`
  - `CabWidgets::LocomotiveSpeed`, 592
- `_up1`
  - `CabWidgets::LocomotiveSpeed`, 590
- `_up10`
  - `CabWidgets::LocomotiveSpeed`, 591, 592
- `_value`
  - `OvalWidgets::OvalScale`, 718
  - `OvalWidgets::OvalSlider`, 723
- `_warrow`
  - `ScrollTabNotebook`, 876
- `_wpage`
  - `ScrollTabNotebook`, 876
- `_writememory`
  - `lcc::ConfigMemory`, 308
  - `lcc::ConfigurationEditor`, 334

- ScrollTabNotebook, [869](#)
- ~AnalogClock
  - Instruments::AnalogClock, [168](#)
- ~Azatrax
  - azatrax::Azatrax, [174](#)
- ~BezierBodyElt
  - Parsers::BezierBodyElt, [188](#)
- ~CMri
  - cmri::CMri, [278](#)
- ~CTCLabel
  - CTCPanel::CTCLabel, [363](#)
- ~CTIAcela
  - CTIAcela, [61](#)
- ~Cab
  - TTSupport::Cab, [193](#)
- ~CabSignalLamp
  - Instruments::CabSignalLamp, [197](#)
- ~CarType
  - FCFSupport::CarType, [270](#)
- ~CatalogDictionary
  - FCFSupport::PDFFileStructures::CatalogDictionary, [273](#)
- ~CodeButton
  - CTCPanel::CodeButton, [292](#)
- ~CornuBodyElt
  - Parsers::CornuBodyElt, [345](#)
- ~CrossReferenceTable
  - FCFSupport::PDFFileStructures::CrossReferenceTable, [359](#)
- ~Crossing
  - CTCPanel::Crossing, [351](#)
- ~Crossover
  - CTCPanel::Crossover, [355](#)
- ~CurvedBlock
  - CTCPanel::CurvedBlock, [396](#)
- ~DialInstrument
  - Instruments::DialInstrument, [401](#)
- ~Dictionary
  - FCFSupport::PDFFileStructures::Dictionary, [403](#)
- ~DigitalClock
  - Instruments::DigitalClock, [406](#)
- ~DigitalInstrument
  - Instruments::DigitalInstrument, [408](#)
- ~Division
  - FCFSupport::Division, [411](#)
- ~DoubleSlip
  - CTCPanel::DoubleSlip, [426](#)
- ~EndBumper
  - CTCPanel::EndBumper, [432](#)
- ~FontDictionary
  - FCFSupport::PDFFileStructures::FontDictionary, [444](#)
- ~FreedObject
  - FCFSupport::PDFFileStructures::FreedObject, [447](#)
- ~GpioInputActiveHigh
  - linuxgpio::GpioInputActiveHigh, [453](#)
- ~GpioInputActiveLow
  - linuxgpio::GpioInputActiveLow, [455](#)
- ~GpioOutputSafeHigh
  - linuxgpio::GpioOutputSafeHigh, [457](#)
- ~GpioOutputSafeHighInvert
  - linuxgpio::GpioOutputSafeHighInvert, [459](#)
- ~GpioOutputSafeLow
  - linuxgpio::GpioOutputSafeLow, [462](#)
- ~GpioOutputSafeLowInverted
  - linuxgpio::GpioOutputSafeLowInverted, [464](#)
- ~HiddenBlock
  - CTCPanel::HiddenBlock, [481](#)
- ~IndirectFloatVector
  - FCFSupport::PDFFileStructures::IndirectFloatVector, [514](#)
- ~IndirectObject
  - FCFSupport::PDFFileStructures::IndirectObject, [517](#)
- ~IndirectObjectDictionary
  - FCFSupport::PDFFileStructures::IndirectObjectDictionary, [522](#)
- ~InformationDirectory
  - FCFSupport::PDFFileStructures::InformationDirectory, [540](#)
- ~LQ24PrinterDevice
  - FCFSupport::LQ24PrinterDevice, [599](#)
- ~Lamp
  - CTCPanel::Lamp, [548](#)
- ~LayoutFile
  - Parsers::LayoutFile, [553](#)
- ~LinuxGpio
  - linuxgpio::LinuxGpio, [571](#)
- ~LogMessageCallback
  - FCFSupport::LogMessageCallback, [596](#)
- ~MRD
  - azatrax::MRD, [623](#)
- ~MRRXtrkCad
  - Parsers::MRRXtrkCad, [632](#)
  - YY\_MRRXtrkCad\_INHERIT, [1278](#)
- ~NCE
  - nce::NCE, [649](#)
- ~OvalButton
  - OvalWidgets::OvalButton, [711](#)
- ~OvalLabel
  - OvalWidgets, [140](#)
- ~OvalRoundCornerRectangle
  - OvalWidgets::OvalRoundCornerRectangle, [713](#)
- ~OvalScale
  - OvalWidgets::OvalScale, [716](#)
- ~OvalScrollBar
  - OvalWidgets::OvalScrollBar, [725](#)
- ~Owner
  - FCFSupport::Owner, [732](#)

- ~PDFNameArray
  - FCFSupport::PDFFileStructures::PDFNameArray, [777](#)
- ~PDFPrinterDevice
  - FCFSupport::PDFPrinterDevice, [780](#)
- ~PDFStream
  - FCFSupport::PDFFileStructures::PDFStream, [787](#)
- ~Page
  - FCFSupport::PDFFileStructures::Page, [736](#)
- ~PageLabelDictionary
  - FCFSupport::PDFFileStructures::PageLabel↔Dictionary, [740](#)
- ~PageLabelTree
  - FCFSupport::PDFFileStructures::PageLabelTree, [743](#)
- ~PageTree
  - FCFSupport::PDFFileStructures::PageTree, [747](#)
- ~ParseFile
  - Parsers::ParseFile, [751](#)
- ~PathName
  - FCFSupport::PathName, [761](#)
  - TTSupport::PathName, [769](#)
- ~PauseCallback
  - FCFSupport::PauseCallback, [775](#)
- ~PostScriptPrinterDevice
  - FCFSupport::PostScriptPrinterDevice, [794](#)
- ~PostScriptStandardType1FontDictionary
  - FCFSupport::PDFFileStructures::PostScript↔StandardType1FontDictionary, [800](#)
- ~PrinterDevice
  - FCFSupport::PrinterDevice, [804](#)
- ~PushButton
  - CTCPanel::PushButton, [812](#)
- ~RaildriverClient
  - raildriver::RaildriverClient, [815](#)
- ~RaildriverIO
  - RaildriverIO, [823](#)
- ~Rectangle
  - FCFSupport::PDFFileStructures::Rectangle, [841](#)
- ~ResourceDictionary
  - FCFSupport::PDFFileStructures::ResourceDictionary, [846](#)
- ~SIGPlate
  - CTCPanel::SIGPlate, [899](#)
- ~SL2
  - azatrax::SL2, [916](#)
- ~SR4
  - azatrax::SR4, [931](#)
- ~SWPlate
  - CTCPanel::SWPlate, [1021](#)
- ~Satellite
  - Satellite, [853](#)
- ~SchLabel
  - CTCPanel::SchLabel, [856](#)
- ~ScissorCrossover
  - CTCPanel::ScissorCrossover, [860](#)
- ~ShowBannerCallback
  - FCFSupport::ShowBannerCallback, [891](#)
- ~Signal
  - CTCPanel::Signal, [894](#)
- ~SingleSlip
  - CTCPanel::SingleSlip, [911](#)
- ~Station
  - FCFSupport::Station, [943](#)
  - TTSupport::Station, [949](#)
- ~Stop
  - TTSupport::Stop, [978](#)
- ~StorageTrack
  - TTSupport::StorageTrack, [986](#)
- ~StraightBlock
  - CTCPanel::StraightBlock, [995](#)
- ~StubYard
  - CTCPanel::StubYard, [999](#)
- ~Switch
  - CTCPanel::Switch, [1003](#)
- ~SwitchList
  - FCFSupport::SwitchList, [1007](#)
- ~System
  - FCFSupport::System, [1037](#)
- ~TextPrinterDevice
  - FCFSupport::TextPrinterDevice, [1112](#)
- ~ThreeWaySW
  - CTCPanel::ThreeWaySW, [1118](#)
- ~ThroughYard
  - CTCPanel::ThroughYard, [1122](#)
- ~TimeTableSystem
  - TTSupport::TimeTableSystem, [1134](#)
- ~Toggle
  - CTCPanel::Toggle, [1156](#)
- ~TrackBody
  - Parsers::TrackBody, [1161](#)
- ~TrackBodyElt
  - Parsers::TrackBodyElt, [1165](#)
- ~TrackGraph
  - Parsers::TrackGraph, [1176](#)
- ~Train
  - FCFSupport::Train, [1210](#)
- ~TrainDisplayCallback
  - FCFSupport::TrainDisplayCallback, [1221](#)
- ~TurnoutBodyElt
  - Parsers::TurnoutBodyElt, [1233](#)
- ~Type1FontDictionary
  - FCFSupport::PDFFileStructures::Type1Font↔Dictionary, [1245](#)
- ~TypedDictionary
  - FCFSupport::PDFFileStructures::TypedDictionary, [1248](#)
- ~WorkInProgressCallback

- FCFSupport::WorkInProgressCallback, [1250](#)
- ~XPressNet
  - xpressnet::XPressNet, [1256](#)
- ~XpressNetEvent
  - xpressnet::XpressNetEvent, [1274](#)
- a
  - HTMLHelp::HTMLHelp, [489](#)
  - Parsers::TrackBodyElt, [1166](#)
  - Parsers::TrackGraph::EdgeValues, [429](#)
- ADDRESSP\_MASK
  - lcc::MTIDetail, [638](#)
- ADDRESSP\_SHIFT
  - lcc::MTIDetail, [638](#)
- AbstractMRMessage
  - lcc, [126](#)
- AbstractMessage
  - lcc, [125](#)
- AccessoryDecoderInformation
  - xpressnet::AccessoryDecoderInformation, [162](#)
- AccessoryDecoderInformationRequest
  - xpressnet::XPressNet, [1259](#)
- AccessoryDecoderOperation
  - nce::NCE, [651](#)
  - xpressnet::XPressNet, [1259](#)
- AccessoryNumber
  - nce, [129](#)
- AccessoryType
  - xpressnet::AccessoryDecoderInformation, [162](#)
- Activate
  - CTIAcela, [47](#)
- add
  - ScrollTabNotebook, [870](#)
- AddCab
  - TTSupport::TimeTableSystem, [1134](#)
- AddCar
  - FCFSupport::System, [1037](#)
- AddColorSpace
  - FCFSupport::PDFFileStructures::ResourceDictionary, [846](#)
- AddExternalGraphicsState
  - FCFSupport::PDFFileStructures::ResourceDictionary, [846](#)
- AddFont
  - FCFSupport::PDFFileStructures::ResourceDictionary, [847](#)
- AddIndirectObject
  - FCFSupport::PDFFileStructures::IndirectObject↔Dictionary, [522](#)
- AddIndirectObjectToTable
  - FCFSupport::PDFFileStructures::CrossReference↔Table, [359](#)
- AddLeadLocomotiveToMultiUnit
  - nce::NCE, [651](#)
- AddLocomotiveToMultiUnit
  - nce::NCE, [652](#)
  - xpressnet::XPressNet, [1260](#)
- AddNewNode
  - Parsers::TrackGraph, [1176](#)
- AddNote
  - TTSupport::Stop, [979](#)
  - TTSupport::TimeTableSystem, [1135](#)
- AddNoteToStop
  - TTSupport::Train, [1198](#)
- AddNoteToTrain
  - TTSupport::Train, [1198](#)
- AddOwner
  - FCFSupport::System, [1038](#)
- AddPage
  - FCFSupport::PDFFileStructures::CatalogDictionary, [274](#)
  - FCFSupport::PDFFileStructures::PageTree, [747](#)
- AddPageLabelDictionary
  - FCFSupport::PDFFileStructures::CatalogDictionary, [274](#)
  - FCFSupport::PDFFileStructures::PageLabelTree, [743](#)
- AddPageLabelTree
  - FCFSupport::PDFFileStructures::CatalogDictionary, [274](#)
  - FCFSupport::PDFFileStructures::PageLabelTree, [743](#)
- AddPageTree
  - FCFSupport::PDFFileStructures::CatalogDictionary, [275](#)
  - FCFSupport::PDFFileStructures::PageTree, [748](#)
- AddPattern
  - FCFSupport::PDFFileStructures::ResourceDictionary, [847](#)
- AddProcSet
  - FCFSupport::PDFFileStructures::ResourceDictionary, [847](#)
- AddProperties
  - FCFSupport::PDFFileStructures::ResourceDictionary, [848](#)
- AddRearLocomotiveToMultiUnit
  - nce::NCE, [652](#)
- AddShading
  - FCFSupport::PDFFileStructures::ResourceDictionary, [848](#)
- AddStation
  - TTSupport::TimeTableSystem, [1135](#)
- AddStorageTrack
  - TTSupport::Station, [949](#)
  - TTSupport::TimeTableSystem, [1135](#)
- AddSwitchListElement
  - FCFSupport::SwitchList, [1007](#), [1008](#)
- AddTrain



- TTSupport::TimeTableSystem, [1136](#)
- AddTrainLongVersion
  - TTSupport::TimeTableSystem, [1136](#)
- AddXObject
  - FCFSupport::PDFFileStructures::ResourceDictionary, [848](#)
- addchild
  - SimpleDOMEElement, [904](#)
- Address
  - xpressnet::AccessoryDecoderInformation, [164](#)
  - xpressnet::DoubleHeaderInformation, [417](#)
  - xpressnet::LI101XPressNetAddress, [569](#)
  - xpressnet::LocomotiveAddress, [575](#)
  - xpressnet::LocomotiveInformation, [582](#)
- address
  - lcc::ConfigMemory, [310](#)
- Address2
  - xpressnet::DoubleHeaderInformation, [417](#)
  - xpressnet::LocomotiveInformation, [582](#)
- AddressCode
  - cmri::CMri, [281](#)
- AddressInquiryNextMUMember
  - xpressnet::XPressNet, [1260](#)
- AddressInquiryNextMU
  - xpressnet::XPressNet, [1260](#)
- AddressInquiryNextStack
  - xpressnet::XPressNet, [1261](#)
- AddressInquiryPreviousMUMember
  - xpressnet::XPressNet, [1261](#)
- AddressInquiryPreviousMU
  - xpressnet::XPressNet, [1261](#)
- AddressInquiryPreviousStack
  - xpressnet::XPressNet, [1261](#)
- AddressType
  - xpressnet::LocomotiveAddress, [575](#)
- addresstype
  - CTIAcela, [45](#)
- aliasMap
  - lcc::CANGridConnect, [213](#)
- AllConnectedDevices
  - azatrax::Azatrax, [174](#)
- allowExternalChanges
  - azatrax::MRD::status2\_union, [966](#)
- AllowingExternalChanges
  - azatrax::MRD, [624](#)
- AnalogClock
  - Instruments::AnalogClock, [167](#)
- ang0
  - Parsers::BezierBodyElt, [190](#)
  - Parsers::CornuBodyElt, [347](#)
  - Parsers::SegVector, [881](#)
  - Parsers::TurnoutBodyElt, [1237](#)
- ang1
  - Parsers::BezierBodyElt, [190](#)
- Parsers::CornuBodyElt, [347](#)
- Parsers::SegVector, [881](#)
- Parsers::TurnoutBodyElt, [1237](#)
- Angle
  - Parsers::LayoutFile, [553](#)
  - Parsers::TrackGraph, [1176](#)
- angle
  - BezierBody.h, [1301](#)
  - CornuBody.h, [1302](#)
  - Parsers::SegVector, [881](#)
  - Parsers::TrackGraph::NodeValues, [675](#)
  - TurnoutBody.h, [1314](#)
- AppendIndustry
  - FCFSupport::Station, [943](#)
- AppendStation
  - FCFSupport::Division, [411](#)
- AppendStream
  - FCFSupport::PDFFileStructures::Page, [736](#)
- AppendTrackBodyElt
  - Parsers::TrackBody, [1162](#)
- Apply
  - Parsers::TrackGraph::Transform2D, [1225](#)
- Area
  - FCFSupport::Division, [411](#)
- area
  - FCFSupport::Division, [413](#)
- Arrival
  - TTSupport::StationTimes, [957](#)
- arrival
  - TTSupport::StationTimes, [958](#)
- asctime\_r
  - PDFPrinterSupport.h, [1295](#)
- aspect
  - CTCPanel::Signal, [897](#)
- AspectBits
  - nce, [130](#)
- aspectlist
  - Parsers::TrackGraph::NodeValues, [675](#)
- AssignLen
  - FCFSupport::Industry, [529](#)
- assignLen
  - FCFSupport::Industry, [534](#)
- AssignLoco
  - nce::NCE, [652](#)
- Assignments
  - FCFSupport::Car, [247](#)
- assignments
  - FCFSupport::Car, [259](#)
- attribute
  - SimpleDOMEElement, [905](#)
- author
  - FCFSupport::PDFFileStructures::Information←  
Directory, [540](#)
- AutoBrake



- RaildriverIO, [835](#)
- Automatic
  - xpressnet, [158](#)
- Available
  - xpressnet::DoubleHeaderInformation, [418](#)
  - xpressnet::LocomotiveInformation, [583](#)
- available
  - lcc::ConfigOptions, [314](#)
- Azatrax, [23](#)
  - azatrax::Azatrax, [174](#)
  - azatrax::MRD, [629](#)
  - azatrax::SL2, [921](#)
  - azatrax::SR4, [937](#)
- azatrax, [75](#)
- Azatrax.h
  - ErrorCode, [1285](#)
  - stopwatchFract, [1286](#)
  - stopwatchSeconds, [1286](#)
- azatrax::Azatrax, [168](#)
  - ~Azatrax, [174](#)
  - AllConnectedDevices, [174](#)
  - Azatrax, [174](#)
  - commands, [171](#)
  - DeviceConnectionList, [173](#)
  - deviceOpenCount, [180](#)
  - GetProductId, [175](#)
  - GetStateData, [175](#)
  - handle, [181](#)
  - Identify\_1, [175](#)
  - IsThisTheAzatraxWeAreLookingFor, [176](#)
  - MRD, [180](#)
  - MyProduct, [176](#)
  - MyProductId, [176](#)
  - myProductId, [181](#)
  - mySerialNumber, [181](#)
  - NumberOfOpenDevices, [177](#)
  - OpenDevice, [177](#)
  - PacketCount, [177](#)
  - ProductIdCode, [178](#)
  - RestoreLEDFunction, [178](#)
  - SL2, [180](#)
  - SR4, [180](#)
  - send2Bytes, [178](#)
  - send3Bytes, [179](#)
  - sendByte, [179](#)
  - SerialNumber, [180](#)
  - stateDataPacket, [181](#)
- azatrax::Azatrax::StateDataPacket, [937](#)
  - commandEcho, [938](#)
  - endOfData, [938](#)
  - operatingMode, [939](#)
  - packetCount, [939](#)
  - reserved, [939](#)
  - status1, [939](#)
  - status2, [939](#)
  - status3, [940](#)
  - status4, [940](#)
  - stopwatchHours, [940](#)
  - stopwatchMinutes, [940](#)
- azatrax::MRD::status1\_union, [963](#)
  - latch\_1, [963](#)
  - latch\_2, [964](#)
  - modtype, [964](#)
  - reserved, [964](#)
  - sense\_1, [964](#)
  - sense\_2, [964](#)
  - theBits, [965](#)
  - theByte, [965](#)
- azatrax::MRD::status2\_union, [965](#)
  - allowExternalChanges, [966](#)
  - externallyChanged, [966](#)
  - reserved, [966](#)
  - resetStatus, [966](#)
  - stopwatchTicking, [966](#)
  - theBits, [967](#)
  - theByte, [967](#)
- azatrax::MRD, [621](#)
  - ~MRD, [623](#)
  - AllowingExternalChanges, [624](#)
  - Azatrax, [629](#)
  - ClearExternallyChanged, [624](#)
  - DisableExternal, [624](#)
  - EnableExternal, [624](#)
  - ExternallyChanged, [625](#)
  - HasRelays, [625](#)
  - Identify\_1\_2, [625](#)
  - Identify\_2, [625](#)
  - Latch\_1, [626](#)
  - Latch\_2, [626](#)
  - MRD, [623](#)
  - OperatingMode, [626](#)
  - OperatingMode\_Type, [623](#)
  - ResetStatus, [626](#)
  - ResetStopwatch, [627](#)
  - Sense\_1, [627](#)
  - Sense\_2, [627](#)
  - SetChan1, [627](#)
  - SetChan2, [628](#)
  - Stopwatch, [628](#)
  - StopwatchTicking, [628](#)
- azatrax::SL2, [913](#)
  - ~SL2, [916](#)
  - Azatrax, [921](#)
  - Input\_1\_Enabled, [916](#)
  - Input\_2\_Enabled, [916](#)
  - Input\_3\_Enabled, [916](#)
  - Input\_4\_Enabled, [916](#)
  - Motor\_1\_Direction, [917](#)

- Motor\_1\_State, [917](#)
- Motor\_2\_Direction, [917](#)
- Motor\_2\_State, [917](#)
- OutputRelayInputControl, [918](#)
- SL2, [915](#)
- Sense\_1, [918](#)
- Sense\_2, [918](#)
- Sense\_3, [919](#)
- Sense\_4, [919](#)
- SetQ1Q2open, [920](#)
- SetQ1negQ2pos, [919](#)
- SetQ1posQ2neg, [919](#)
- SetQ3Q4open, [920](#)
- SetQ3negQ4pos, [920](#)
- SetQ3posQ4neg, [920](#)
- azatrax::SL2::status1\_union, [961](#)
  - motor\_1\_direction, [961](#)
  - motor\_1\_state, [961](#)
  - motor\_2\_direction, [962](#)
  - motor\_2\_state, [962](#)
  - reserved, [962](#)
  - theBits, [962](#)
  - theByte, [962](#)
- azatrax::SL2::status2\_union, [969](#)
  - reserved, [970](#)
  - sense\_1, [970](#)
  - sense\_2, [970](#)
  - sense\_3, [970](#)
  - sense\_4, [970](#)
  - theBits, [971](#)
  - theByte, [971](#)
- azatrax::SL2::status3\_union, [973](#)
  - input\_1\_enabled, [974](#)
  - input\_2\_enabled, [974](#)
  - input\_3\_enabled, [974](#)
  - input\_4\_enabled, [974](#)
  - reserved, [974](#)
  - theBits, [975](#)
  - theByte, [975](#)
- azatrax::SR4, [928](#)
  - ~SR4, [931](#)
  - Azatrax, [937](#)
  - BlinkRelays, [931](#)
  - Input\_1\_Enabled, [931](#)
  - Input\_2\_Enabled, [932](#)
  - Input\_3\_Enabled, [932](#)
  - Input\_4\_Enabled, [932](#)
  - OutputRelayInputControl, [932](#)
  - PulseRelays, [933](#)
  - Q1\_State, [933](#)
  - Q2\_State, [933](#)
  - Q3\_State, [934](#)
  - Q4\_State, [934](#)
  - RelaysOff, [934](#)
  - RelaysOn, [935](#)
  - SR4, [930](#)
  - Sense\_1\_Latch, [935](#)
  - Sense\_1\_Live, [935](#)
  - Sense\_2\_Latch, [935](#)
  - Sense\_2\_Live, [936](#)
  - Sense\_3\_Latch, [936](#)
  - Sense\_3\_Live, [936](#)
  - Sense\_4\_Latch, [936](#)
  - Sense\_4\_Live, [937](#)
- azatrax::SR4::status1\_union, [958](#)
  - Q1\_state, [959](#)
  - Q2\_state, [959](#)
  - Q3\_state, [959](#)
  - Q4\_state, [960](#)
  - reserved, [960](#)
  - theBits, [960](#)
  - theByte, [960](#)
- azatrax::SR4::status2\_union, [967](#)
  - reserved, [968](#)
  - sense\_1, [968](#)
  - sense\_2, [968](#)
  - sense\_3, [968](#)
  - sense\_4, [968](#)
  - theBits, [969](#)
  - theByte, [969](#)
- azatrax::SR4::status3\_union, [971](#)
  - input\_1\_enabled, [972](#)
  - input\_2\_enabled, [972](#)
  - input\_3\_enabled, [972](#)
  - input\_4\_enabled, [972](#)
  - reserved, [972](#)
  - theBits, [973](#)
  - theByte, [973](#)
- BISON\_YYLTYPE\_ISDECLARED
  - MRRXtrkCad.tab.h, [1305](#)
- back
  - HTMLHelp::HTMLHelp, [489](#)
- backcurrenttopic
  - HTMLHelp::HTMLHelp, [489](#)
- backpointers
  - Parsers::TrackGraph, [1193](#)
- BailOff
  - RaildriverIO, [835](#)
- bar
  - CabWidgets::LocomotiveSpeed, [592](#)
- base
  - ScrollTabNotebook, [876](#)
- baseFont
  - FCFSupport::PDFFileStructures::Type1Font↔
    - Dictionary, [1245](#)
- basepin
  - linuxgpio::GpioInputActiveHigh, [454](#)

- linuxgpio::GpioInputActiveLow, [456](#)
- linuxgpio::GpioOutputSafeHigh, [458](#)
- linuxgpio::GpioOutputSafeHighInvert, [460](#)
- linuxgpio::GpioOutputSafeLow, [462](#)
- linuxgpio::GpioOutputSafeLowInverted, [465](#)
- basicFormatCheck
  - lcc::GridConnectReply, [475](#)
- BezierBody
  - Parsers::BezierBody, [183](#)
  - Parsers::BezierBodyElt, [190](#)
- BezierBody.h
  - angle, [1301](#)
  - len0, [1301](#)
  - len1, [1301](#)
- BezierBodyElt
  - Parsers::BezierBody, [185](#)
  - Parsers::BezierBodyElt, [188](#)
- BezierBodyEltType
  - Parsers::BezierBodyElt, [187](#)
- BezierEnds
  - Parsers::BezierBody, [183](#)
- BezierSegmentCount
  - Parsers::BezierBody, [183](#)
- bind
  - CTCPanel::CTCPanel, [372](#)
  - FileEntry, [96](#)
  - LabelComboBox, [106](#)
  - LabelSpinBox, [116](#)
- bitstype
  - lcc::OpenLCBProtocols, [709](#)
- Blink
  - CTIAcela, [47](#)
- BlinkRelays
  - azatrax::SR4, [931](#)
- bufferize
  - Parsers::ParseFile, [752](#)
- buildPortandnidDialog
  - lcc::CANGridConnectOverUSBSerial, [227](#)
- buildPortnidandhostDialog
  - lcc::CANGridConnectOverTcp, [222](#)
  - lcc::OpenLCBOverTcp, [703](#)
- buildSocketnamenidDialog
  - lcc::CANGridConnectOverCANSocket, [218](#)
- buttons
  - lcc::ConfigurationEditor, [337](#)
  - mainwindow, [619](#)
- buttons\_add
  - mainwindow, [609](#)
- buttons\_delete
  - mainwindow, [609](#)
- buttons\_hide
  - mainwindow, [609](#)
- buttons\_insert
  - mainwindow, [609](#)
- buttons\_itemconfigure
  - mainwindow, [610](#)
- buttons\_show
  - mainwindow, [610](#)
- byte
  - lcc, [122](#)
- ByteList
  - cmri, [77](#)
- bytelist
  - lcc, [122](#)
- bytelist72
  - lcc, [122](#)
- C++/Azatrax/Azatrax.h, [1285](#)
- C++/Azatrax/mrd.h, [1286](#)
- C++/Azatrax/sl2.h, [1286](#)
- C++/Azatrax/sr4.h, [1287](#)
- C++/FCFSupport/CallBack.h, [1287](#)
- C++/FCFSupport/Car.h, [1288](#)
- C++/FCFSupport/CarType.h, [1288](#)
- C++/FCFSupport/Division.h, [1289](#)
- C++/FCFSupport/FCFSupportGroup.h, [1290](#)
- C++/FCFSupport/Industry.h, [1290](#)
- C++/FCFSupport/LQ24Printer.h, [1290](#)
- C++/FCFSupport/Owner.h, [1291](#)
- C++/FCFSupport/PDFPrinter.h, [1292](#)
- C++/FCFSupport/PDFPrinterSupport.h, [1293](#)
- C++/FCFSupport/PathName.h, [1291](#)
- C++/FCFSupport/PostScriptPrinter.h, [1295](#)
- C++/FCFSupport/Printer.h, [1296](#)
- C++/FCFSupport/Station.h, [1296](#)
- C++/FCFSupport/SwitchList.h, [1297](#)
- C++/FCFSupport/System.h, [1298](#)
- C++/FCFSupport/TextPrinter.h, [1299](#)
- C++/FCFSupport/Train.h, [1299](#)
- C++/ParserClasses/BezierBody.h, [1300](#)
- C++/ParserClasses/CornuBody.h, [1302](#)
- C++/ParserClasses/IntegerList.h, [1303](#)
- C++/ParserClasses/MRRXtrkCad.tab.h, [1303](#)
- C++/ParserClasses/ParseFile.h, [1311](#)
- C++/ParserClasses/ParserClassesGroup.h, [1311](#)
- C++/ParserClasses/SocketPair.h, [1312](#)
- C++/ParserClasses/TrackBody.h, [1312](#)
- C++/ParserClasses/TrackGraph.h, [1312](#)
- C++/ParserClasses/TurnoutBody.h, [1313](#)
- C++/RailDriver/RaildriverIO.h, [1314](#)
- C++/TTSupport/Cab.h, [1315](#)
- C++/TTSupport/PathName.h, [1292](#)
- C++/TTSupport/Station.h, [1297](#)
- C++/TTSupport/TTSupportGroup.h, [1318](#)
- C++/TTSupport/TimeTableSystem.h, [1316](#)
- C++/TTSupport/TimeTableSystemTcl.h, [1318](#)
- C++/TTSupport/Train.h, [1300](#)
- C++/TclSocketCAN/TclSocketCAN.i, [1315](#)

- C++/wiringPi/tclwiringpi.i, [1318](#)
- c\_idMap
  - Parsers::TrackGraph, [1194](#)
- c\_nodes
  - Parsers::TrackGraph, [1194](#)
- c\_roots
  - Parsers::TrackGraph, [1194](#)
- CANGridConnect
  - lcc::CANGridConnect, [205](#)
- CANGridConnectOverCANSocket
  - lcc::CANGridConnectOverCANSocket, [217](#)
- CANGridConnectOverTcp
  - lcc::CANGridConnectOverTcp, [221](#)
- CANGridConnectOverUSBSerial
  - lcc::CANGridConnectOverUSBSerial, [226](#)
- CANHeader
  - lcc::CANHeader, [230](#)
- CENTIMETERSperMM
  - MRRXtrkCad.tab.h, [1309](#)
- CMri
  - cmri::CMri, [278](#)
- CPData
  - CTCPanel::CTCPanel, [389](#)
- CPList
  - CTCPanel::CTCPanel, [389](#)
- CSAddress
  - nce, [130](#)
- CTCLabel
  - CTCPanel::CTCLabel, [363](#)
- CTCPanel, [79](#)
  - CTCPanel::CTCPanel, [369](#)
  - leverMethods, [81](#)
  - standardMethods, [81](#)
  - trackworkmethods, [82](#)
  - verifyBoolMethod, [82](#)
  - verifyColorMethod, [82](#)
  - verifyDoubleMethod, [82](#)
  - verifyOrientation8Method, [82](#)
  - verifyPositionMethod, [82](#)
- CTCPanel::CTCLabel, [361](#)
  - \_configureColor, [363](#)
  - \_configureLabel, [364](#)
  - ~CTCLabel, [363](#)
  - CTCLabel, [363](#)
  - canvas, [365](#)
  - ctcpanel, [365](#)
  - geti, [364](#)
  - getv, [364](#)
  - invoke, [364](#)
  - seti, [364](#)
  - setv, [365](#)
- CTCPanel::CTCPanel, [365](#)
  - \_CtcMainHScroll2, [371](#)
  - \_CtcMainSyncX, [371](#)
  - \_PosInteger, [372](#)
  - \_ch\_oldfocus, [388](#)
  - \_ch\_oldgrab, [388](#)
  - \_crosshairEnd, [370](#)
  - \_crosshairMove, [370](#)
  - \_crosshairStart, [371](#)
  - bind, [372](#)
  - CPData, [389](#)
  - CPList, [389](#)
  - CTCPanel, [369](#)
  - checkInitCP, [372](#)
  - controls, [388](#)
  - controls\_crosshair, [373](#)
  - controlsYscroll, [388](#)
  - coords, [373](#)
  - cplist, [374](#)
  - create\_CTCLabel, [375](#)
  - create\_CodeButton, [374](#)
  - create\_Crossing, [374](#)
  - create\_Crossover, [374](#)
  - create\_CurvedBlock, [375](#)
  - create\_DoubleSlip, [375](#)
  - create\_EndBumper, [376](#)
  - create\_HiddenBlock, [376](#)
  - create\_Lamp, [376](#)
  - create\_PushButton, [377](#)
  - create\_SIGPlate, [378](#)
  - create\_SWPlate, [380](#)
  - create\_SchLabel, [377](#)
  - create\_ScissorCrossover, [377](#)
  - create\_Signal, [378](#)
  - create\_SingleSlip, [378](#)
  - create\_StraightBlock, [379](#)
  - create\_StubYard, [379](#)
  - create\_Switch, [379](#)
  - create\_ThreeWaySW, [380](#)
  - create\_ThroughYard, [380](#)
  - create\_Toggle, [381](#)
  - delete, [381](#)
  - exists, [381](#)
  - getZoom, [382](#)
  - geti, [382](#)
  - getv, [382](#)
  - invoke, [382](#)
  - itemcget, [383](#)
  - itemconfigure, [383](#)
  - lappendCP, [383](#)
  - lremoveCP, [384](#)
  - move, [384](#)
  - objectlist, [385](#)
  - Objects, [389](#)
  - print, [385](#)
  - scale, [389](#)
  - schematic, [389](#)

- schematic\_crosshair, 385
- schematicYscroll, 389
- setZoom, 386
- seti, 386
- setv, 386
- updateAndSyncCP, 387
- updateSR, 387
- xscroll, 390
- zoomBy, 388
- CTCPanel::CodeButton, 291
  - ~CodeButton, 292
  - canvas, 293
  - CodeButton, 292
  - ctcpanel, 293
  - geti, 292
  - getv, 292
  - invoke, 293
  - seti, 293
  - setv, 293
- CTCPanel::Crossing, 349
  - \_VerifyCrossingType, 351
  - \_configureLabel, 351
  - ~Crossing, 351
  - canvas, 353
  - Crossing, 351
  - ctcpanel, 353
  - geti, 352
  - getv, 352
  - invoke, 352
  - seti, 352
  - setv, 352
- CTCPanel::Crossover, 353
  - \_configureLabel, 355
  - ~Crossover, 355
  - canvas, 357
  - Crossover, 355
  - ctcpanel, 357
  - geti, 356
  - getv, 356
  - invoke, 356
  - seti, 356
  - setv, 356
  - state, 357
- CTCPanel::CurvedBlock, 394
  - \_PI, 398
  - \_RadiansToDegrees, 396
  - \_configureLabel, 396
  - \_square, 397
  - ~CurvedBlock, 396
  - canvas, 398
  - ctcpanel, 398
  - CurvedBlock, 396
  - geti, 397
  - invoke, 397
  - seti, 397
  - setv, 397
- CTCPanel::DoubleSlip, 423
  - \_configureLabel, 426
  - ~DoubleSlip, 426
  - canvas, 427
  - ctcpanel, 427
  - DoubleSlip, 425
  - geti, 426
  - getv, 426
  - invoke, 426
  - seti, 427
  - setv, 427
  - state, 428
- CTCPanel::EndBumper, 430
  - \_configureLabel, 432
  - ~EndBumper, 432
  - canvas, 433
  - ctcpanel, 433
  - EndBumper, 431
  - geti, 432
  - invoke, 432
  - seti, 432
  - setv, 433
- CTCPanel::HiddenBlock, 479
  - \_configureLabel, 481
  - ~HiddenBlock, 481
  - canvas, 483
  - ctcpanel, 483
  - geti, 481
  - getv, 482
  - HiddenBlock, 481
  - invoke, 482
  - seti, 482
  - setv, 482
- CTCPanel::Lamp, 546
  - \_configureColor, 548
  - \_configureLabel, 548
  - ~Lamp, 548
  - canvas, 549
  - ctcpanel, 550
  - geti, 548
  - getv, 548
  - invoke, 549
  - Lamp, 547
  - seti, 549
  - setv, 549
  - state, 550
- CTCPanel::PushButton, 810
  - \_configureLabel, 812
  - ~PushButton, 812
  - canvas, 813
  - ctcpanel, 813
  - geti, 812

- getv, [812](#)
- invoke, [812](#)
- PushButton, [811](#)
- seti, [812](#)
- setv, [813](#)
- CTCPanel::SIGPlate, [897](#)
  - \_PlatePolygon, [901](#)
  - \_configureLabel, [900](#)
  - ~SIGPlate, [899](#)
  - canvas, [901](#)
  - ctcpanel, [901](#)
  - geti, [900](#)
  - getv, [900](#)
  - invoke, [900](#)
  - SIGPlate, [899](#)
  - seti, [900](#)
  - setv, [901](#)
- CTCPanel::SWPlate, [1019](#)
  - \_PlatePolygon, [1023](#)
  - \_configureLabel, [1021](#)
  - ~SWPlate, [1021](#)
  - canvas, [1023](#)
  - ctcpanel, [1023](#)
  - geti, [1021](#)
  - getv, [1022](#)
  - invoke, [1022](#)
  - SWPlate, [1021](#)
  - seti, [1022](#)
  - setv, [1022](#)
- CTCPanel::SchLabel, [854](#)
  - \_configureColor, [856](#)
  - \_configureLabel, [856](#)
  - ~SchLabel, [856](#)
  - canvas, [858](#)
  - ctcpanel, [858](#)
  - geti, [857](#)
  - getv, [857](#)
  - invoke, [857](#)
  - SchLabel, [856](#)
  - seti, [857](#)
  - setv, [857](#)
- CTCPanel::ScissorCrossover, [858](#)
  - \_configureLabel, [860](#)
  - ~ScissorCrossover, [860](#)
  - canvas, [862](#)
  - ctcpanel, [862](#)
  - geti, [861](#)
  - getv, [861](#)
  - invoke, [861](#)
  - ScissorCrossover, [860](#)
  - seti, [861](#)
  - setv, [861](#)
  - state, [862](#)
- CTCPanel::Signal, [892](#)
  - \_SchematicDrawOval, [894](#)
  - \_SchematicDrawThinLine, [895](#)
  - \_VerifyHeads, [895](#)
  - \_configureLabel, [894](#)
  - ~Signal, [894](#)
  - aspect, [897](#)
  - canvas, [897](#)
  - ctcpanel, [897](#)
  - geti, [895](#)
  - getv, [896](#)
  - invoke, [896](#)
  - seti, [896](#)
  - setv, [896](#)
  - Signal, [893](#)
- CTCPanel::SingleSlip, [909](#)
  - \_configureLabel, [911](#)
  - ~SingleSlip, [911](#)
  - canvas, [913](#)
  - ctcpanel, [913](#)
  - geti, [912](#)
  - getv, [912](#)
  - invoke, [912](#)
  - seti, [912](#)
  - setv, [912](#)
  - SingleSlip, [911](#)
  - state, [913](#)
- CTCPanel::StraightBlock, [992](#)
  - \_configureLabel, [995](#)
  - ~StraightBlock, [995](#)
  - canvas, [996](#)
  - ctcpanel, [996](#)
  - geti, [995](#)
  - getv, [995](#)
  - invoke, [995](#)
  - seti, [996](#)
  - setv, [996](#)
  - StraightBlock, [994](#)
- CTCPanel::StubYard, [997](#)
  - \_StubYard\_Poly, [1000](#)
  - \_configureLabel, [999](#)
  - ~StubYard, [999](#)
  - canvas, [1000](#)
  - ctcpanel, [1001](#)
  - geti, [999](#)
  - getv, [999](#)
  - invoke, [999](#)
  - seti, [1000](#)
  - setv, [1000](#)
  - StubYard, [998](#)
- CTCPanel::Switch, [1001](#)
  - \_configureLabel, [1003](#)
  - ~Switch, [1003](#)
  - canvas, [1005](#)
  - ctcpanel, [1005](#)

- geti, [1003](#)
- getv, [1004](#)
- invoke, [1004](#)
- seti, [1004](#)
- setv, [1004](#)
- state, [1005](#)
- Switch, [1003](#)
- CTCPanel::ThreeWaySW, [1115](#)
  - \_configureLabel, [1118](#)
  - ~ThreeWaySW, [1118](#)
  - canvas, [1119](#)
  - ctcpanel, [1119](#)
  - geti, [1118](#)
  - getv, [1118](#)
  - invoke, [1118](#)
  - seti, [1119](#)
  - setv, [1119](#)
  - state, [1120](#)
  - ThreeWaySW, [1117](#)
- CTCPanel::ThroughYard, [1120](#)
  - \_ThroughYard\_Poly, [1123](#)
  - \_configureLabel, [1122](#)
  - ~ThroughYard, [1122](#)
  - canvas, [1123](#)
  - ctcpanel, [1124](#)
  - geti, [1122](#)
  - invoke, [1122](#)
  - seti, [1123](#)
  - setv, [1123](#)
  - ThroughYard, [1121](#)
- CTCPanel::Toggle, [1154](#)
  - \_AddTLever, [1156](#)
  - \_MoveTLever, [1157](#)
  - \_VerifyOrientationHV, [1158](#)
  - \_configureCenterLabel, [1157](#)
  - \_configureLeftLabel, [1157](#)
  - \_configureRightLabel, [1157](#)
  - ~Toggle, [1156](#)
  - canvas, [1160](#)
  - ctcpanel, [1160](#)
  - geti, [1158](#)
  - getv, [1158](#)
  - invoke, [1158](#)
  - lever, [1160](#)
  - seti, [1158](#)
  - setv, [1159](#)
  - Toggle, [1156](#)
- CTI\_DeviceMap
  - CTIAcela, [61](#)
- CTIAcela, [42](#)
  - \_handleSRQ, [46](#)
  - \_readbyte, [46](#)
  - \_readevent, [47](#)
  - \_timeout, [61](#)
  - \_transmit, [47](#)
  - ~CTIAcela, [61](#)
  - Activate, [47](#)
  - addresstype, [45](#)
  - Blink, [47](#)
  - CTI\_DeviceMap, [61](#)
  - CTIAcela, [50](#)
  - ConfigureSensor, [48](#)
  - Control16, [48](#)
  - Control4, [49](#)
  - Control8, [50](#)
  - ctiacela, [61](#)
  - dataavailable, [61](#)
  - Deactive, [51](#)
  - EmergencyStop, [51](#)
  - FilterSelectBits, [61](#)
  - filterthreshtype, [45](#)
  - HaveData, [51](#)
  - highbyte, [51](#)
  - LampBits, [62](#)
  - lowbyte, [52](#)
  - maxtries, [62](#)
  - momtype, [45](#)
  - NetworkOffline, [52](#)
  - NetworkOnline, [52](#)
  - networkonline, [62](#)
  - OnlineP, [52](#)
  - Opcodes, [62](#)
  - pack4, [53](#)
  - pack8, [53](#)
  - Poll, [54](#)
  - PulseOff, [54](#)
  - PulseOn, [54](#)
  - Query, [55](#)
  - Read, [55](#)
  - Read16, [55](#)
  - Read4, [56](#)
  - Read8, [56](#)
  - ReadAll, [57](#)
  - ReadRevision, [57](#)
  - ResetNetwork, [57](#)
  - Responses, [62](#)
  - ReverseBlink, [57](#)
  - SRQControl, [60](#)
  - Signal2, [58](#)
  - Signal3, [58](#)
  - Signal4, [58](#)
  - SignalBrightness, [59](#)
  - SignalSettings, [59](#)
  - speedtype, [45](#)
  - Throttle, [60](#)
  - ttyfd, [62](#)
  - ubyte, [46](#)
  - validate, [60](#)

- Cab, [37](#)
  - CabNameMap, [37](#)
  - TTSupport::Cab, [193](#)
- cab
  - TTSupport::Stop, [983](#)
- CabNameMap
  - Cab, [37](#)
- CabNumber
  - nce, [130](#)
- CabSignalLamp
  - Instruments::CabSignalLamp, [197](#)
- CabWidgets, [76](#)
- CabWidgets::LocomotiveDirection, [575](#)
  - \_direction, [578](#)
  - \_left, [578](#)
  - \_right, [579](#)
  - \_setdirection, [577](#)
  - currentDirection, [579](#)
  - direction, [577](#)
  - direction\_sense, [578](#)
  - forward, [579](#)
  - invoke, [578](#)
  - LocomotiveDirection, [577](#)
  - reverse, [579](#)
- CabWidgets::LocomotiveSpeed, [587](#)
  - \_down, [591](#)
  - \_down1, [590](#)
  - \_down10, [590](#), [592](#)
  - \_setspeed, [590](#)
  - \_speed, [592](#)
  - \_stop, [590](#), [592](#)
  - \_up, [592](#)
  - \_up1, [590](#)
  - \_up10, [591](#), [592](#)
  - bar, [592](#)
  - down1, [593](#)
  - down10, [593](#)
  - invoke, [591](#)
  - leftbuttons, [593](#)
  - LocomotiveSpeed, [589](#)
  - rightbuttons, [593](#)
  - setspeed, [591](#)
  - speed, [591](#)
  - stop, [593](#)
  - up1, [593](#)
  - up10, [594](#)
- CabWidgets::SelectLocomotive, [883](#)
  - \_addnewloco, [885](#)
  - \_trimList, [885](#)
  - currentLocomotive, [885](#)
  - invoke, [885](#)
  - lf, [886](#)
  - locoList, [886](#)
  - SelectLocomotive, [884](#)
- cabs
  - TTSupport::TimeTableSystem, [1152](#)
- CanAlias
  - lcc::CanAlias, [201](#)
- CanMessage
  - lcc::CanMessage, [235](#)
- CanTransport
  - lcc::CanTransport, [240](#)
- canheader
  - lcc::CANGridConnect, [213](#)
  - lcc::MTIHeader, [644](#)
- canvas
  - CTCPanel::CTCLabel, [365](#)
  - CTCPanel::CodeButton, [293](#)
  - CTCPanel::Crossing, [353](#)
  - CTCPanel::Crossover, [357](#)
  - CTCPanel::CurvedBlock, [398](#)
  - CTCPanel::DoubleSlip, [427](#)
  - CTCPanel::EndBumper, [433](#)
  - CTCPanel::HiddenBlock, [483](#)
  - CTCPanel::Lamp, [549](#)
  - CTCPanel::PushButton, [813](#)
  - CTCPanel::SIGPlate, [901](#)
  - CTCPanel::SWPlate, [1023](#)
  - CTCPanel::SchLabel, [858](#)
  - CTCPanel::ScissorCrossover, [862](#)
  - CTCPanel::Signal, [897](#)
  - CTCPanel::SingleSlip, [913](#)
  - CTCPanel::StraightBlock, [996](#)
  - CTCPanel::StubYard, [1000](#)
  - CTCPanel::Switch, [1005](#)
  - CTCPanel::ThreeWaySW, [1119](#)
  - CTCPanel::ThroughYard, [1123](#)
  - CTCPanel::Toggle, [1160](#)
  - Instruments::CabSignalLamp, [199](#)
  - OvalWidgets, [140](#)
  - OvalWidgets::OvalButton, [712](#)
  - OvalWidgets::OvalRoundCornerRectangle, [713](#)
  - OvalWidgets::OvalScale, [718](#)
  - OvalWidgets::OvalSlider, [723](#)
  - OvalWidgets::OvalScrollBar, [729](#)
- Car
  - FCFSupport::Car, [245](#), [246](#)
- CarAssignment
  - FCFSupport::System, [1038](#)
- carDest
  - FCFSupport::System, [1098](#)
- CarGroup
  - FCFSupport::CarGroup, [265](#)
- CarGroupConsts
  - FCFSupport::CarGroup, [264](#)
- carGroups
  - FCFSupport::System, [1098](#)
- CarLocationType



- FCFSupport::System, [1036](#)
- CarMovements
  - FCFSupport::System, [1038](#)
- carMovements
  - FCFSupport::System, [1099](#)
- CarOwner
  - FCFSupport::Car, [247](#)
- CarType
  - FCFSupport::CarType, [269](#)
- CarTypeConsts
  - FCFSupport::CarType, [268](#)
- CarTypeMap
  - FCFSupport, [86](#)
- CarTypeOrderVector
  - FCFSupport, [86](#)
- CarTypeReport
  - FCFSupport::System, [1036](#)
- CarTypes
  - FCFSupport::Train, [1211](#)
- carTypes
  - FCFSupport::System, [1101](#)
  - FCFSupport::Train, [1217](#)
- CarTypesFile
  - FCFSupport::System, [1041](#)
- carTypesFile
  - FCFSupport::System, [1101](#)
- CarTypesOrder
  - FCFSupport::System, [1041](#)
- carTypesOrder
  - FCFSupport::System, [1102](#)
- CarTypesOrderIndex
  - FCFSupport::System, [1041](#)
- CarVector
  - FCFSupport, [86](#)
- CardType
  - cmri, [78](#)
- CardType\_Byte
  - cmri::CMri, [281](#)
- cars
  - FCFSupport::Industry, [534](#)
  - FCFSupport::System, [1099](#)
- CarsAtDest
  - FCFSupport::System, [1039](#)
- carsAtDest
  - FCFSupport::System, [1099](#)
- CarsAtDest\_CarsInTransit
  - FCFSupport::System, [1039](#)
- carsAtDest\_carsInTransit
  - FCFSupport::System, [1099](#)
- CarsAtWorkBench
  - FCFSupport::System, [1039](#)
- carsAtWorkBench
  - FCFSupport::System, [1099](#)
- CarsFile
  - FCFSupport::System, [1039](#)
- carsFile
  - FCFSupport::System, [1100](#)
- CarsInTransit
  - FCFSupport::System, [1039](#)
- carsInTransit
  - FCFSupport::System, [1100](#)
- CarsLen
  - FCFSupport::Industry, [529](#)
- carsLen
  - FCFSupport::Industry, [534](#)
- CarsMoved
  - FCFSupport::System, [1040](#)
- carsMoved
  - FCFSupport::System, [1100](#)
- CarsMovedMore
  - FCFSupport::System, [1040](#)
- carsMovedMore
  - FCFSupport::System, [1100](#)
- CarsMovedOnce
  - FCFSupport::System, [1040](#)
- carsMovedOnce
  - FCFSupport::System, [1100](#)
- CarsMovedThree
  - FCFSupport::System, [1040](#)
- carsMovedThree
  - FCFSupport::System, [1101](#)
- CarsMovedTwice
  - FCFSupport::System, [1040](#)
- carsMovedTwice
  - FCFSupport::System, [1101](#)
- CarsNotMoved
  - FCFSupport::System, [1041](#)
- carsNotMoved
  - FCFSupport::System, [1101](#)
- CarsNum
  - FCFSupport::Industry, [529](#)
- carsNum
  - FCFSupport::Industry, [534](#)
- CatalogDictionary
  - FCFSupport::PDFFileStructures::CatalogDictionary, [273](#)
- cdi
  - lcc::ConfigurationEditor, [337](#)
- cget
  - FileEntry, [96](#)
  - LabelComboBox, [107](#)
  - LabelSelectColor, [112](#)
  - LabelSpinBox, [117](#)
- ChCodes
  - FCFSupport::LQ24PrinterDevice, [598](#)
- ChangeMomentumLevel
  - nce::NCE, [653](#)
- CheckColor

- splash, [925](#)
- CheckForResponse
  - xpressnet::XPressNet, [1262](#)
- CheckImage
  - splash, [925](#)
- checkInitCP
  - CTCPanel::CTCPanel, [372](#)
- children
  - SimpleDOMEElement, [905](#)
- circleLayoutP
  - Parsers::TrackGraph, [1194](#)
- ClassNumber
  - TTSupport::Train, [1199](#)
- classnumber
  - TTSupport::Train, [1204](#)
- CleanUpBezierBody
  - Parsers::BezierBody, [184](#)
- CleanUpCornuBody
  - Parsers::CornuBody, [340](#)
- CleanUpElement
  - Parsers::BezierBody, [184](#)
  - Parsers::CornuBody, [340](#)
  - Parsers::TurnoutBody, [1228](#)
- CleanUpIntegerList
  - Parsers::IntegerList, [543](#)
- CleanUpTurnoutBody
  - Parsers::TurnoutBody, [1228](#)
- Cleanup
  - Parsers::TrackGraph::NodeValues, [675](#)
- clear
  - raildriver::RaildriverClient, [815](#)
- ClearAssignments
  - FCFSupport::Car, [247](#)
- ClearExternallyChanged
  - azatrax::MRD, [624](#)
- ClearMovementsThisSession
  - FCFSupport::Car, [247](#)
- ClearTrips
  - FCFSupport::Car, [248](#)
- ClosePrinter
  - FCFSupport::LQ24PrinterDevice, [600](#)
  - FCFSupport::PDFPrinterDevice, [780](#)
  - FCFSupport::PostScriptPrinterDevice, [794](#)
  - FCFSupport::PrinterDevice, [804](#)
  - FCFSupport::TextPrinterDevice, [1113](#)
- CloseTrainDisplay
  - FCFSupport::TrainDisplayCallback, [1221](#)
- closeport
  - CmriSupport::CmriNode, [288](#)
- Clr
  - linuxgpio::GpioOutputSafeHighInvert, [460](#)
  - linuxgpio::GpioOutputSafeLowInverted, [464](#)
  - linuxgpio::LinuxGpio, [571](#)
- Cmri, [41](#)
- cmri, [76](#)
  - ByteList, [77](#)
  - CardType, [78](#)
  - uatype, [78](#)
  - ubyte, [78](#)
- Cmri Support code, [68](#)
- cmri::Cmri, [276](#)
  - \_readbyte, [278](#)
  - \_readevent, [279](#)
  - \_timeout, [281](#)
  - \_transmit, [279](#)
  - ~Cmri, [278](#)
  - AddressCode, [281](#)
  - Cmri, [278](#)
  - CardType\_Byte, [281](#)
  - DLE, [281](#)
  - ETX, [282](#)
  - Init, [282](#)
  - InitBoard, [279](#)
  - Inputs, [280](#)
  - Outputs, [281](#)
  - Poll, [282](#)
  - Read, [282](#)
  - STX, [282](#)
  - Transmit, [282](#)
  - ttyfd, [283](#)
- CmriNode
  - CmriSupport::CmriNode, [284](#)
- CmriSupport, [79](#)
- CmriSupport::CmriNode, [283](#)
  - \_TypeCodes, [290](#)
  - \_ValidateAddress, [285](#)
  - \_ValidateByte, [285](#)
  - \_ValidateListOfBytes, [287](#)
  - \_ValidateSixElementListOfBytes, [287](#)
  - \_ValidateType, [287](#)
  - \_ValidateWord, [288](#)
  - closeport, [288](#)
  - CmriNode, [284](#)
  - inputs, [288](#)
  - openport, [288](#)
  - outputbuffer, [290](#)
  - outputs, [289](#)
  - portopenp, [289](#)
  - setbitfield, [289](#)
  - setport, [290](#)
  - validate, [290](#)
- CodeButton
  - CTCPanel::CodeButton, [292](#)
- Color
  - TTSupport::Cab, [194](#)
- color
  - HTMLHelp::HTMLHelp, [490](#)
  - TTSupport::Cab, [195](#)

- ColorFillOption
  - OvalWidgets, [138](#)
- ColorOptionMethods
  - OvalWidgets, [138](#)
- ColorOutlineOption
  - OvalWidgets, [138](#)
- ColorPopup
  - LabelSelectColor, [112](#)
- colorSpace
  - FCFSupport::PDFFileStructures::ResourceDictionary, [849](#)
- command
  - HTMLHelp::HTMLHelp, [509](#)
- commandEcho
  - azatrax::Azatrax::StateDataPacket, [938](#)
- CommandStationResponse
  - xpressnet::CommandStationResponse, [295](#)
- CommandStationSoftwareVersion
  - xpressnet::XPressNet, [1262](#)
- CommandStationStatus
  - xpressnet::CommandStationStatus, [297](#)
- CommandStationStatusRequest
  - xpressnet::XPressNet, [1262](#)
- CommandStationTypeCode
  - xpressnet::SoftwareVersion, [922](#)
- commands
  - azatrax::Azatrax, [171](#)
- Comment
  - FCFSupport::CarType, [270](#)
  - FCFSupport::Owner, [732](#)
  - FCFSupport::Station, [944](#)
- comment
  - FCFSupport::CarType, [271](#)
  - FCFSupport::Owner, [733](#)
  - FCFSupport::Station, [945](#)
- CommonOptions
  - Instruments, [103](#)
- CommonValidateMethods
  - OvalWidgets, [139](#)
- complete
  - lcc, [125](#)
- Completed
  - xpressnet::AccessoryDecoderInformation, [164](#)
- CompressGraph
  - Parsers::LayoutFile, [556](#)
  - Parsers::TrackGraph, [1179](#)
- compressed\_edge\_exists
  - Parsers::TrackGraph, [1177](#)
- CompressedEdgeCount
  - Parsers::LayoutFile, [554](#)
  - Parsers::TrackGraph, [1177](#)
- CompressedEdgeLength
  - Parsers::LayoutFile, [554](#)
  - Parsers::TrackGraph, [1177](#)
- CompressedEdgeNode
  - Parsers::LayoutFile, [554](#)
  - Parsers::TrackGraph, [1177](#)
- CompressedEdgePair
  - Parsers::TrackGraph, [1173](#)
- CompressedEdgePairVector
  - Parsers::TrackGraph, [1173](#)
- CompressedEdgeValues
  - Parsers::TrackGraph::CompressedEdgeValues, [300](#)
- CompressedGraph
  - Parsers::TrackGraph, [1173](#)
- CompressedGraphCircleLayout
  - Parsers::LayoutFile, [554](#)
  - Parsers::TrackGraph, [1177](#)
- CompressedGraphKamadaKawaiSpring
  - Parsers::LayoutFile, [555](#)
  - Parsers::TrackGraph, [1178](#)
- CompressedGraphKruskalMinimumSpanningTree
  - Parsers::LayoutFile, [555](#)
  - Parsers::TrackGraph, [1178](#)
- CompressedGraphPrimMinimumSpanningTree
  - Parsers::LayoutFile, [555](#)
  - Parsers::TrackGraph, [1178](#)
- CompressedIdNodeMap
  - Parsers::TrackGraph, [1174](#)
- CompressedNode
  - Parsers::TrackGraph, [1174](#)
- CompressedNodePositionX
  - Parsers::LayoutFile, [555](#)
  - Parsers::TrackGraph, [1178](#)
- CompressedNodePositionY
  - Parsers::LayoutFile, [556](#)
  - Parsers::TrackGraph, [1179](#)
- CompressedNodeSegments
  - Parsers::LayoutFile, [556](#)
  - Parsers::TrackGraph, [1179](#)
- CompressedNodeValues
  - Parsers::TrackGraph::CompressedNodeValues, [301](#)
- CompressedNodeVector
  - Parsers::TrackGraph, [1174](#)
- compressedP
  - Parsers::TrackGraph, [1194](#)
- compute\_size
  - ScrollTabNotebook, [870](#)
- computeHeads
  - Parsers::TrackGraph, [1179](#)
- ComputeRouteLength
  - Parsers::TrackGraph, [1179](#)
- ComputeTimes
  - TTSupport::TimeTableSystem, [1137](#)
- ConcatCornuBody
  - Parsers::CornuBody, [340](#)
- ConfigMemory
  - lcc::ConfigMemory, [305](#)

- ConfigOptions
  - lcc::ConfigOptions, [312](#)
- ConfigurationEditor
  - lcc::ConfigurationEditor, [318](#)
- ConfigurationType
  - ReadConfiguration, [146](#)
- configure
  - FileEntry, [96](#)
  - LabelComboBox, [107](#)
  - LabelSelectColor, [114](#)
  - LabelSpinBox, [117](#)
- ConfigureSensor
  - CTIAcela, [48](#)
- ConnectedTrackEnd
  - Parsers::TrackBodyElt, [1165](#)
- ConsBezierBody
  - Parsers::BezierBody, [184](#)
- ConsCornuBody
  - Parsers::CornuBody, [341](#)
- ConsTrackBody
  - Parsers::TrackBody, [1162](#)
- ConsTurnoutBody
  - Parsers::TurnoutBody, [1229](#)
- ConsistAddress
  - nce, [130](#)
  - xpressnet, [155](#)
- constructorCombo
  - lcc::OpenLCBNode, [699](#)
- ConsumerIdentified
  - lcc::OpenLCBNode, [691](#)
- ConsumerRangeIdentified
  - lcc::OpenLCBNode, [692](#)
- ContainsTime
  - TTSupport::TimeRange, [1126](#)
- contents
  - FCFSupport::PDFFileStructures::Page, [737](#)
- Control16
  - CTIAcela, [48](#)
- Control4
  - CTIAcela, [49](#)
- Control8
  - CTIAcela, [50](#)
- controls
  - CTCPanel::CTCPanel, [388](#)
- controls\_crosshair
  - CTCPanel::CTCPanel, [373](#)
- controlsYscroll
  - CTCPanel::CTCPanel, [388](#)
- coords
  - CTCPanel::CTCPanel, [373](#)
- copy
  - lcc::CanMessage, [236](#)
- CopyList
  - Parsers::IntegerList, [543](#)
- CornuBody
  - Parsers::CornuBody, [340](#)
  - Parsers::CornuBodyElt, [347](#)
- CornuBody.h
  - angle, [1302](#)
  - len0, [1302](#)
  - len1, [1302](#)
- CornuBodyElt
  - Parsers::CornuBody, [342](#)
  - Parsers::CornuBodyElt, [345](#)
- CornuBodyEltType
  - Parsers::CornuBodyElt, [344](#)
- CornuEnds
  - Parsers::CornuBody, [341](#)
- CornuSegmentCount
  - Parsers::CornuBody, [341](#)
- count
  - lcc::ConfigMemory, [310](#)
- countNUL
  - lcc::CANGridConnect, [207](#)
- cplist
  - CTCPanel::CTCPanel, [374](#)
- create
  - FileEntry, [97](#)
  - LabelComboBox, [107](#)
  - LabelSelectColor, [114](#)
  - LabelSpinBox, [117](#)
- create\_CTCLabel
  - CTCPanel::CTCPanel, [375](#)
- create\_CodeButton
  - CTCPanel::CTCPanel, [374](#)
- create\_Crossing
  - CTCPanel::CTCPanel, [374](#)
- create\_Crossover
  - CTCPanel::CTCPanel, [374](#)
- create\_CurvedBlock
  - CTCPanel::CTCPanel, [375](#)
- create\_DoubleSlip
  - CTCPanel::CTCPanel, [375](#)
- create\_EndBumper
  - CTCPanel::CTCPanel, [376](#)
- create\_HiddenBlock
  - CTCPanel::CTCPanel, [376](#)
- create\_Lamp
  - CTCPanel::CTCPanel, [376](#)
- create\_PushButton
  - CTCPanel::CTCPanel, [377](#)
- create\_SIGPlate
  - CTCPanel::CTCPanel, [378](#)
- create\_SWPlate
  - CTCPanel::CTCPanel, [380](#)
- create\_SchLabel
  - CTCPanel::CTCPanel, [377](#)
- create\_ScissorCrossover

- CTCPanel::CTCPanel, [377](#)
- create\_Signal
  - CTCPanel::CTCPanel, [378](#)
- create\_SingleSlip
  - CTCPanel::CTCPanel, [378](#)
- create\_StraightBlock
  - CTCPanel::CTCPanel, [379](#)
- create\_StubYard
  - CTCPanel::CTCPanel, [379](#)
- create\_Switch
  - CTCPanel::CTCPanel, [379](#)
- create\_ThreeWaySW
  - CTCPanel::CTCPanel, [380](#)
- create\_ThroughYard
  - CTCPanel::CTCPanel, [380](#)
- create\_Toggle
  - CTCPanel::CTCPanel, [381](#)
- CreateLaTeXTimetable
  - TTSupport::TimeTableSystem, [1137](#)
- CreateNewPage
  - FCFSupport::PDFPrinterDevice, [780](#)
- CreateNewStream
  - FCFSupport::PDFPrinterDevice, [780](#)
- createReply
  - lcc::GridConnectReply, [475](#)
- creator
  - FCFSupport::PDFFileStructures::Information↔  
Directory, [540](#)
- creationDate
  - FCFSupport::PDFFileStructures::Information↔  
Directory, [541](#)
- cropBox
  - FCFSupport::PDFFileStructures::Page, [737](#)
  - FCFSupport::PDFFileStructures::PageTree, [748](#)
- CrossReferenceTable
  - FCFSupport::PDFFileStructures::CrossReference↔  
Table, [359](#)
  - FCFSupport::PDFFileStructures::IndirectObject, [520](#)
- crossReferenceTable
  - FCFSupport::PDFPrinterDevice, [783](#)
- Crossing
  - CTCPanel::Crossing, [351](#)
- Crossover
  - CTCPanel::Crossover, [355](#)
- ctcpanel
  - CTCPanel::CTCLabel, [365](#)
  - CTCPanel::CodeButton, [293](#)
  - CTCPanel::Crossing, [353](#)
  - CTCPanel::Crossover, [357](#)
  - CTCPanel::CurvedBlock, [398](#)
  - CTCPanel::DoubleSlip, [427](#)
  - CTCPanel::EndBumper, [433](#)
  - CTCPanel::HiddenBlock, [483](#)
  - CTCPanel::Lamp, [550](#)
  - CTCPanel::PushButton, [813](#)
  - CTCPanel::SIGPlate, [901](#)
  - CTCPanel::SWPlate, [1023](#)
  - CTCPanel::SchLabel, [858](#)
  - CTCPanel::ScissorCrossover, [862](#)
  - CTCPanel::Signal, [897](#)
  - CTCPanel::SingleSlip, [913](#)
  - CTCPanel::StraightBlock, [996](#)
  - CTCPanel::StubYard, [1001](#)
  - CTCPanel::Switch, [1005](#)
  - CTCPanel::ThreeWaySW, [1119](#)
  - CTCPanel::ThroughYard, [1124](#)
  - CTCPanel::Toggle, [1160](#)
- ctiacela, [83](#)
  - CTIAcela, [61](#)
- ctiacela::CTIAcela, [390](#)
- curDiv
  - FCFSupport::System, [1102](#)
- currentColumn
  - FCFSupport::LQ24PrinterDevice, [603](#)
  - FCFSupport::PDFPrinterDevice, [783](#)
  - FCFSupport::TextPrinterDevice, [1115](#)
- currentColumnFraction
  - FCFSupport::LQ24PrinterDevice, [603](#)
  - FCFSupport::PDFPrinterDevice, [784](#)
- currentDirection
  - CabWidgets::LocomotiveDirection, [579](#)
- currentFontName
  - FCFSupport::PDFPrinterDevice, [784](#)
- currentLocomotive
  - CabWidgets::SelectLocomotive, [885](#)
- currentPage
  - FCFSupport::PDFPrinterDevice, [784](#)
- currentProgress
  - splash, [927](#)
- CurrentScale
  - Parsers::MRRXtrkCad, [633](#)
  - YY\_MRRXtrkCad\_INHERIT, [1279](#)
- currentSlant
  - FCFSupport::LQ24PrinterDevice, [603](#)
- currentSpacing
  - FCFSupport::LQ24PrinterDevice, [603](#)
- currentStream
  - FCFSupport::PDFPrinterDevice, [784](#)
- currentWeight
  - FCFSupport::LQ24PrinterDevice, [604](#)
- curtopicindex
  - HTMLHelp::HTMLHelp, [509](#)
- CurvedBlock
  - CTCPanel::CurvedBlock, [396](#)
- CV
  - xpressnet::ServiceModeResponse, [889](#)
- DESTID\_MASK

- lcc::MTIDetail, [638](#)
- DESTID\_SHIFT
  - lcc::MTIDetail, [639](#)
- DIRECTIONFMT
  - linuxgpio::LinuxGpio, [573](#)
- DLE
  - cmri::CMri, [281](#)
- dTextX
  - Instruments::DialInstrument, [401](#)
- dTextY
  - Instruments::DialInstrument, [402](#)
- Data
  - xpressnet::ServiceModeResponse, [889](#)
- data
  - SimpleDOMEElement, [905](#)
- dataavailable
  - CTIAcela, [61](#)
- databuf
  - lcc, [122](#)
- DatagramReceivedOK
  - lcc::OpenLCBNode, [692](#)
- DatagramRejected
  - lcc::OpenLCBNode, [692](#)
- datagrambuffer
  - lcc::ConfigMemory, [310](#)
  - lcc::ConfigurationEditor, [337](#)
- datagrambuffers
  - lcc::CANGridConnect, [213](#)
  - lcc::OpenLCBOverTcp, [705](#)
- datagramcontent
  - lcc, [125](#)
- dbg
  - ScrollTabNotebook, [876](#)
- Deactive
  - CTIAcela, [51](#)
- DecoderLongAddress
  - xpressnet, [155](#)
- defaultHelpDirectory
  - HTMLHelp::HTMLHelp, [509](#)
- defaultHelpWindow
  - HTMLHelp::HTMLHelp, [509](#)
- defaultTableOfContents
  - HTMLHelp::HTMLHelp, [510](#)
- DegreesToRadians
  - GRSupport, [100](#)
- delete
  - CTCPanel::CTCPanel, [381](#)
- DeleteAllExistingCars
  - FCFSupport::System, [1042](#)
- DeleteLocomotiveFromStack
  - xpressnet::XPressNet, [1262](#)
- DeleteTrain
  - TTSupport::TimeTableSystem, [1138](#)
- DeleteTurnoutGraphic
  - Parsers::TrackGraph, [1180](#)
- DeleteTurnoutRouteList
  - Parsers::TrackGraph, [1180](#)
- deliver
  - FCFSupport::System, [1102](#)
- delta
  - OvalWidgets::OvalScrollBar, [727](#)
- Departure
  - TTSupport::StationTimes, [957](#)
  - TTSupport::Stop, [979](#)
  - TTSupport::Train, [1199](#)
- departure
  - TTSupport::StationTimes, [958](#)
  - TTSupport::Train, [1204](#)
- DequeuePacket
  - nce::NCE, [653](#)
- Description
  - FCFSupport::CarGroup, [266](#)
  - FCFSupport::Train, [1211](#)
- description
  - FCFSupport::CarGroup, [267](#)
  - FCFSupport::Train, [1217](#)
- Destination
  - FCFSupport::Car, [248](#)
- destination
  - FCFSupport::Car, [259](#)
- Determinant
  - Parsers::TrackGraph::Transform2D, [1225](#)
- DeviceConnectionList
  - azatrax::Azatrax, [173](#)
- deviceOpenCount
  - azatrax::Azatrax, [180](#)
- DialInstrument
  - Instruments::DialInstrument, [400](#)
- Dictionary
  - FCFSupport::PDFFileStructures::Dictionary, [403](#)
- Digital1
  - RaildriverIO, [836](#)
- Digital2
  - RaildriverIO, [836](#)
- Digital3
  - RaildriverIO, [836](#)
- Digital4
  - RaildriverIO, [836](#)
- Digital5
  - RaildriverIO, [836](#)
- Digital6
  - RaildriverIO, [836](#)
- DigitalClock
  - Instruments::DigitalClock, [405](#)
- DigitalInstrument
  - Instruments::DigitalInstrument, [408](#)
- DirectModeCVRead
  - xpressnet::XPressNet, [1263](#)

- DirectModeCVWrite
  - xpressnet::XPressNet, [1263](#)
- Direction
  - nce, [133](#)
  - xpressnet::DoubleHeaderInformation, [418](#)
  - xpressnet::LocomotiveInformation, [583](#)
- direction
  - CabWidgets::LocomotiveDirection, [577](#)
- direction\_sense
  - CabWidgets::LocomotiveDirection, [578](#)
- DirectionCode
  - xpressnet, [157](#)
- DirectionName
  - TTSupport::TimeTableSystem, [1152](#)
- Dirname
  - FCFSupport::PathName, [761](#)
  - TTSupport::PathName, [769](#)
- DisableExternal
  - azatrax::MRD, [624](#)
- DisableMain
  - nce::NCE, [654](#)
- DiscardSwitchList
  - FCFSupport::SwitchList, [1008](#)
- display
  - SimpleDOMEElement, [905](#)
- displayTree
  - ParseXML, [757](#)
- DissolveDoubleHeader
  - xpressnet::XPressNet, [1263](#)
- divList
  - FCFSupport::Train, [1217](#)
- Division
  - FCFSupport::Division, [410](#)
- division
  - FCFSupport::Station, [945](#)
- DivisionControlList
  - FCFSupport::Industry, [529](#)
- divisionControlList
  - FCFSupport::Industry, [534](#)
- DivisionList
  - FCFSupport::Train, [1211](#)
- DivisionMap
  - FCFSupport, [87](#)
- DivisionSymbolMap
  - FCFSupport, [87](#)
- DivisionVector
  - FCFSupport, [87](#)
- Divisions
  - FCFSupport::Car, [248](#)
- divisions
  - FCFSupport::Car, [259](#)
  - FCFSupport::System, [1102](#)
- Doc/doxygen/titlepage.h, [1320](#)
- Done
  - FCFSupport::Train, [1211](#)
- done
  - FCFSupport::Train, [1217](#)
- doneP
  - FCFSupport::Car, [259](#)
- DoubleHeaderInformation
  - xpressnet::DoubleHeaderInformation, [416](#)
- DoubleHeaderMuError
  - xpressnet::DoubleHeaderMuError, [423](#)
- DoubleSlip
  - CTCPanel::DoubleSlip, [425](#)
- doubleVector
  - TTSupport, [149](#)
- down1
  - CabWidgets::LocomotiveSpeed, [593](#)
- down10
  - CabWidgets::LocomotiveSpeed, [593](#)
- drawOptionsDialog
  - lcc::CANGridConnectOverCANSocket, [218](#)
  - lcc::CANGridConnectOverTcp, [222](#)
  - lcc::CANGridConnectOverUSBSerial, [227](#)
  - lcc::OpenLCBOverTcp, [703](#)
- dropStop
  - FCFSupport::SwitchListElement, [1018](#)
- DropStopEQ
  - FCFSupport::SwitchListElement, [1015](#)
- DropStopIndustry
  - FCFSupport::SwitchListElement, [1016](#)
- DropStopStation
  - FCFSupport::SwitchListElement, [1016](#)
- Dummy
  - nce::NCE, [654](#)
- DuplicateStationIndex
  - TTSupport::Station, [949](#)
  - TTSupport::TimeTableSystem, [1138](#)
- duplicateStationIndex
  - TTSupport::Station, [952](#)
- ePos1
  - Parsers::SegVector, [881](#)
- ePos2
  - Parsers::SegVector, [881](#)
- ETX
  - cmri::CMri, [282](#)
- EVENTIDFMT
  - lcc::EventID, [437](#)
- EVENTP\_MASK
  - lcc::MTIDetail, [639](#)
- EVENTP\_SHIFT
  - lcc::MTIDetail, [639](#)
- EXPORT
  - linuxgpio::LinuxGpio, [573](#)
- EchoMode
  - nce, [130](#)

- EdgeIndex
  - Parsers::LayoutFile, [557](#)
  - Parsers::TrackGraph, [1180](#)
- EdgeLength
  - Parsers::LayoutFile, [557](#)
  - Parsers::TrackGraph, [1180](#)
- EdgeValues
  - Parsers::TrackGraph::EdgeValues, [429](#)
- EdgeA
  - Parsers::LayoutFile, [556](#)
  - Parsers::TrackGraph, [1180](#)
- EdgeX
  - Parsers::LayoutFile, [557](#)
  - Parsers::TrackGraph, [1181](#)
- EdgeY
  - Parsers::LayoutFile, [557](#)
  - Parsers::TrackGraph, [1181](#)
- editframe
  - lcc::ConfigurationEditor, [337](#)
- eightbytes
  - lcc, [123](#)
- Element
  - Parsers::BezierBody, [184](#)
  - Parsers::CornuBody, [341](#)
  - Parsers::IntegerList, [543](#)
  - Parsers::TurnoutBody, [1229](#)
- element
  - Parsers::BezierBody, [185](#)
  - Parsers::CornuBody, [342](#)
  - Parsers::TrackBody, [1163](#)
  - Parsers::TurnoutBody, [1230](#)
- ElementAddress
  - xpressnet, [155](#)
- ElementP
  - Parsers::IntegerList, [544](#)
- elements
  - FCFSupport::PDFFileStructures::IndirectObject↔  
Dictionary, [523](#)
- EmergencyOff
  - xpressnet::CommandStationStatus, [297](#)
- EmergencyStop
  - CTIAcela, [51](#)
  - xpressnet::CommandStationStatus, [297](#)
- EmergencyStopALocomotive
  - xpressnet::XPressNet, [1264](#)
- EmergencyStopAllLocomotives
  - xpressnet::XPressNet, [1264](#)
- Emit
  - Parsers::LayoutFile, [558](#)
- EmptiesAccepted
  - FCFSupport::Industry, [529](#)
- emptyTypes
  - FCFSupport::Industry, [535](#)
- EmptyP
  - FCFSupport::Car, [248](#)
- enableClickDestroy
  - splash, [926](#)
- EnableExternal
  - azatrax::MRD, [624](#)
- EnableMain
  - nce::NCE, [654](#)
- encodingDictionary
  - FCFSupport::PDFFileStructures::Type1Font↔  
Dictionary, [1246](#)
- encodingName
  - FCFSupport::PDFFileStructures::Type1Font↔  
Dictionary, [1246](#)
- EndBumper
  - CTCPanel::EndBumper, [431](#)
- endOfData
  - azatrax::Azatrax::StateDataPacket, [938](#)
- equals
  - lcc::CanMessage, [236](#)
- Error
  - xpressnet::DoubleHeaderMuError, [423](#)
- ErrorCode
  - Azatrax.h, [1285](#)
- ErrorMessage
  - nce, [134](#)
- ErrorCode
  - xpressnet, [157](#)
- errorstream
  - Parsers::ParseFile, [752](#)
- EstablishDoubleHeader
  - xpressnet::XPressNet, [1264](#)
- EventID
  - lcc::EventID, [435](#)
- EventLog
  - lcc::EventLog, [439](#)
- EventReceived
  - lcc::EventReceived, [441](#)
- eventReceived
  - lcc::EventLog, [440](#)
- Eventcodes
  - RaildriverIO, [821](#)
- eventid
  - lcc::EventReceived, [442](#)
  - lcc::SendEvent, [888](#)
- eventlist
  - raildriver, [144](#)
- Eventmask\_bits
  - RaildriverIO, [822](#)
- eventvalidity
  - lcc, [125](#)
- ExecuteMacro
  - nce::NCE, [654](#)
- exists
  - CTCPanel::CTCPanel, [381](#)



extGState  
     FCFSupport::PDFFileStructures::ResourceDictionary, [849](#)  
 Extension  
     FCFSupport::PathName, [761](#)  
     TTSupport::PathName, [769](#)  
 ExternallyChanged  
     azatrax::MRD, [625](#)  
 externallyChanged  
     azatrax::MRD::status2\_union, [966](#)  
  
 FALSE  
     MRRXtrkCad.tab.h, [1305](#)  
 FCFSupport, [73](#), [84](#)  
     CarTypeMap, [86](#)  
     CarTypeOrderVector, [86](#)  
     CarVector, [86](#)  
     DivisionMap, [87](#)  
     DivisionSymbolMap, [87](#)  
     DivisionVector, [87](#)  
     IndustryMap, [87](#)  
     IndustryVector, [87](#)  
     operator<<, [89](#)  
     OwnerMap, [87](#)  
     StationMap, [88](#)  
     StationVector, [88](#)  
     stringVector, [88](#)  
     SwitchListElementVector, [88](#)  
     TrainMap, [88](#)  
     TrainNameMap, [89](#)  
 FCFSupport::Car, [241](#)  
     Assignments, [247](#)  
     assignments, [259](#)  
     Car, [245](#), [246](#)  
     CarOwner, [247](#)  
     ClearAssignments, [247](#)  
     ClearMovementsThisSession, [247](#)  
     ClearTrips, [248](#)  
     Destination, [248](#)  
     destination, [259](#)  
     Divisions, [248](#)  
     divisions, [259](#)  
     doneP, [259](#)  
     EmptyP, [248](#)  
     FixedRouteP, [248](#)  
     fixedP, [260](#)  
     IncrementAssignments, [249](#)  
     IncrementTrips, [249](#)  
     IncrmentMovementsThisSession, [249](#)  
     IsDoneP, [249](#)  
     LastTrain, [249](#)  
     lasttrain, [260](#)  
     LdLmt, [250](#)  
     Idlmt, [260](#)  
     Length, [250](#)  
     length, [260](#)  
     Load, [250](#)  
     LoadedP, [250](#)  
     loadedP, [260](#)  
     Location, [250](#)  
     location, [261](#)  
     LtWt, [251](#)  
     ltwt, [261](#)  
     Marks, [251](#)  
     marks, [261](#)  
     mirrorP, [261](#)  
     MovementsThisSession, [251](#)  
     moves, [261](#)  
     Number, [251](#)  
     number, [262](#)  
     OkToMirrorP, [251](#)  
     operator=, [252](#)  
     owner, [262](#)  
     Peek, [252](#)  
     peek, [262](#)  
     Plate, [252](#)  
     plate, [262](#)  
     PrevTrain, [252](#)  
     prevtrain, [262](#)  
     SetAssignments, [253](#)  
     SetCarOwner, [253](#)  
     SetDestination, [253](#)  
     SetDivisions, [253](#)  
     SetDone, [254](#)  
     SetFixedRouteP, [254](#)  
     SetLastTrain, [254](#)  
     SetLdLmt, [254](#)  
     SetLength, [255](#)  
     SetLocation, [255](#)  
     SetLtWt, [255](#)  
     SetMarks, [255](#)  
     SetNotDone, [256](#)  
     SetNumber, [256](#)  
     SetOkToMirrorP, [256](#)  
     SetPeek, [256](#)  
     SetPlate, [257](#)  
     SetPrevTrain, [257](#)  
     SetType, [257](#)  
     SetWeightClass, [257](#)  
     System, [259](#)  
     tmpStatus, [263](#)  
     Trips, [258](#)  
     trips, [263](#)  
     Type, [258](#)  
     type, [263](#)  
     UnLoad, [258](#)  
     WeightClass, [258](#)  
     weightclass, [263](#)

- FCFSupport::CarGroup, 264
  - CarGroup, 265
  - CarGroupConsts, 264
  - Description, 266
  - description, 267
  - Group, 266
  - group, 267
  - operator=, 266
- FCFSupport::CarType, 267
  - ~CarType, 270
  - CarType, 269
  - CarTypeConsts, 268
  - Comment, 270
  - comment, 271
  - Group, 270
  - group, 271
  - operator=, 270
  - System, 271
  - Type, 271
  - type, 272
- FCFSupport::Division, 409
  - ~Division, 411
  - AppendStation, 411
  - Area, 411
  - area, 413
  - Division, 410
  - Home, 412
  - home, 414
  - Name, 412
  - name, 414
  - NumberOfStations, 412
  - operator=, 412
  - stations, 414
  - Symbol, 413
  - symbol, 414
  - System, 413
  - TheStation, 413
- FCFSupport::Industry, 524
  - AssignLen, 529
  - assignLen, 534
  - cars, 534
  - CarsLen, 529
  - carsLen, 534
  - CarsNum, 529
  - carsNum, 534
  - DivisionControlList, 529
  - divisionControlList, 534
  - EmptiesAccepted, 529
  - emptyTypes, 535
  - Hazard, 530
  - hazard, 535
  - IncrementStatsLen, 530
  - Industry, 526, 528
  - loadTypes, 535
  - LoadsAccepted, 530
  - MaxCarLen, 530
  - maxCarLen, 535
  - MaxPlate, 530
  - MaxWeightClass, 531
  - mirror, 535
  - MyMirror, 531
  - MyStation, 531
  - Name, 531
  - name, 536
  - NumberOfCars, 531
  - operator=, 532
  - plate, 536
  - Priority, 532
  - priority, 536
  - Reload, 532
  - reload, 536
  - remLen, 536
  - station, 537
  - StatsLen, 532
  - statsLen, 537
  - System, 533
  - TheCar, 533
  - TrackLen, 533
  - trackLen, 537
  - Type, 533
  - type, 537
  - usedLen, 537
  - weightclass, 538
- FCFSupport::LQ24PrinterDevice, 597
  - ~LQ24PrinterDevice, 599
  - ChCodes, 598
  - ClosePrinter, 600
  - currentColumn, 603
  - currentColumnFraction, 603
  - currentSlant, 603
  - currentSpacing, 603
  - currentWeight, 604
  - LQ24PrinterDevice, 599
  - NewPage, 600
  - oneColumnWidthFraction, 604
  - OpenPrinter, 600
  - printerStream, 604
  - Put, 601
  - PutLine, 601
  - SetTypeSlant, 601
  - SetTypeSpacing, 602
  - SetTypeWeight, 602
  - Tab, 602
- FCFSupport::LogMessageCallback, 594
  - ~LogMessageCallback, 596
  - LogMessage, 596
  - LogMessageCallback, 596
  - MessageType, 595

- FCFSupport::Owner, 730
  - ~Owner, 732
  - Comment, 732
  - comment, 733
  - Initials, 732
  - initials, 733
  - Name, 732
  - name, 733
  - operator=, 733
  - Owner, 731
- FCFSupport::PDFFileStructures, 90
  - NamedIndirectObjectMap, 91
  - PDFStreamVector, 92
  - PageLabelDictionaryNumMap, 92
  - PageLabelTreeKidVector, 92
  - QuotePDFString, 92
- FCFSupport::PDFFileStructures::CatalogDictionary, 272
  - ~CatalogDictionary, 273
  - AddPage, 274
  - AddPageLabelDictionary, 274
  - AddPageLabelTree, 274
  - AddPageTree, 275
  - CatalogDictionary, 273
  - labels, 275
  - pages, 275
  - WriteDictionaryElements, 275
- FCFSupport::PDFFileStructures::CrossReferenceTable, 357
  - ~CrossReferenceTable, 359
  - AddIndirectObjectToTable, 359
  - CrossReferenceTable, 359
  - FreeObject, 360
  - HighestObjectNumber, 360
  - IndirectObject, 361
  - lastObjectNumber, 361
  - objectMap, 359
  - objectTable, 361
  - WriteTable, 360
- FCFSupport::PDFFileStructures::Dictionary, 402
  - ~Dictionary, 403
  - Dictionary, 403
  - WriteDictionaryElements, 404
  - WriteDirect, 404
- FCFSupport::PDFFileStructures::FontDictionary, 443
  - ~FontDictionary, 444
  - FontDictionary, 444
  - subType, 445
  - WriteDictionaryElements, 444
  - WriteFontType, 445
- FCFSupport::PDFFileStructures::FreedObject, 446
  - ~FreedObject, 447
  - FreedObject, 446
  - WriteDirect, 447
- FCFSupport::PDFFileStructures::IndirectFloatVector, 513
  - ~IndirectFloatVector, 514
  - IndirectFloatVector, 514
  - WriteDirect, 515
- FCFSupport::PDFFileStructures::IndirectObject, 515
  - ~IndirectObject, 517
  - CrossReferenceTable, 520
  - FileOffset, 517
  - fileOffset, 520
  - GenerationNumber, 517
  - generationNumber, 520
  - HasOffset, 517
  - IncrementGenerationNumber, 518
  - IndirectObject, 516
  - ObjectNumber, 518
  - objectNumber, 520
  - SetObjectNumber, 518
  - table, 520
  - WriteDirect, 518
  - WriteIndirectReference, 519
  - WriteObjectToFile, 519
- FCFSupport::PDFFileStructures::IndirectObjectDictionary, 521
  - ~IndirectObjectDictionary, 522
  - AddIndirectObject, 522
  - elements, 523
  - IndirectObjectDictionary, 522
  - Size, 523
  - WriteDictionaryElements, 523
- FCFSupport::PDFFileStructures::InformationDirectory, 538
  - ~InformationDirectory, 540
  - author, 540
  - creator, 540
  - creationDate, 541
  - InformationDirectory, 539
  - keywords, 541
  - modificationDate, 541
  - producer, 541
  - subject, 541
  - title, 541
  - WriteDictionaryElements, 540
- FCFSupport::PDFFileStructures::PDFNameArray, 776
  - ~PDFNameArray, 777
  - PDFNameArray, 777
- FCFSupport::PDFFileStructures::PDFStream, 786
  - ~PDFStream, 787
  - PDFStream, 787
  - WriteDirect, 787
- FCFSupport::PDFFileStructures::Page, 734
  - ~Page, 736
  - AppendStream, 736
  - contents, 737
  - cropBox, 737
  - mediaBox, 737

- Page, 735
- PageTree, 736
- parent, 737
- resources, 737
- WriteDictionaryElements, 736
- FCFSupport::PDFFileStructures::PageLabelDictionary, 738
  - ~PageLabelDictionary, 740
  - NumberStyle, 739
  - PageLabelDictionary, 739
  - prefix, 740
  - start, 741
  - style, 741
  - WriteDictionaryElements, 740
- FCFSupport::PDFFileStructures::PageLabelTree, 741
  - ~PageLabelTree, 743
  - AddPageLabelDictionary, 743
  - AddPageLabelTree, 743
  - GetKidLimits, 744
  - isRoot, 745
  - kids, 745
  - nums, 745
  - PageLabelTree, 742
  - Size, 744
  - WriteDictionaryElements, 744
- FCFSupport::PDFFileStructures::PageTree, 745
  - ~PageTree, 747
  - AddPage, 747
  - AddPageTree, 748
  - cropBox, 748
  - mediaBox, 748
  - PageTree, 746
  - pagenodes, 749
  - parent, 749
  - resources, 749
  - WriteDictionaryElements, 748
- FCFSupport::PDFFileStructures::PostScriptStandardType1FontDictionary, 799
  - ~PostScriptStandardType1FontDictionary, 800
  - PostScriptStandardType1FontDictionary, 800
- FCFSupport::PDFFileStructures::Rectangle, 840
  - ~Rectangle, 841
  - Rectangle, 841
  - WriteDirect, 842
  - X1, 842
  - x1, 843
  - X2, 842
  - x2, 843
  - Y1, 842
  - y1, 843
  - Y2, 842
  - y2, 843
- FCFSupport::PDFFileStructures::ResourceDictionary, 844
  - ~ResourceDictionary, 846
  - AddColorSpace, 846
  - AddExternalGraphicsState, 846
  - AddFont, 847
  - AddPattern, 847
  - AddProcSet, 847
  - AddProperties, 848
  - AddShading, 848
  - AddXObject, 848
  - colorSpace, 849
  - extGState, 849
  - font, 849
  - pattern, 850
  - procSets, 850
  - properties, 850
  - ResourceDictionary, 845
  - shading, 850
  - WriteDictionaryElements, 849
  - xObject, 850
- FCFSupport::PDFFileStructures::Type1FontDictionary, 1242
  - ~Type1FontDictionary, 1245
  - baseFont, 1245
  - encodingDictionary, 1246
  - encodingName, 1246
  - firstChar, 1246
  - fontDescriptor, 1246
  - lastChar, 1246
  - Type1FontDictionary, 1243, 1244
  - widths, 1246
  - WriteDictionaryElements, 1245
- FCFSupport::PDFFileStructures::TypedDictionary, 1247
  - ~TypedDictionary, 1248
  - type, 1249
  - TypedDictionary, 1248
  - WriteDictionaryElements, 1248
  - WriteDictionaryType, 1249
- FCFSupport::PDFPrinterDevice, 777
  - ~PDFPrinterDevice, 780
  - ClosePrinter, 780
  - CreateNewPage, 780
  - CreateNewStream, 780
  - crossReferenceTable, 783
  - currentColumn, 783
  - currentColumnFraction, 784
  - currentFontName, 784
  - currentPage, 784
  - currentStream, 784
  - horizontalScaling, 784
  - info, 784
  - lines, 785
  - maxLines, 785
  - needPage, 785
  - NewPage, 781

- OpenPrinter, [781](#)
- PDFPrinterDevice, [779](#)
- pageTreeRoot, [785](#)
- partline, [785](#)
- printerStream, [785](#)
- Put, [781](#)
- PutLine, [782](#)
- rootDictionary, [786](#)
- SetTypeSlant, [782](#)
- SetTypeSpacing, [782](#)
- SetTypeWeight, [783](#)
- Tab, [783](#)
- title, [786](#)
- FCFSupport::PathName, [758](#)
  - ~PathName, [761](#)
  - Dirname, [761](#)
  - Extension, [761](#)
  - FullPath, [761](#)
  - operator<, [763](#)
  - operator<=, [763](#)
  - operator>, [764](#)
  - operator>=, [765](#)
  - operator+, [761](#), [762](#)
  - operator+=, [762](#)
  - operator=, [763](#), [764](#)
  - operator==, [764](#)
  - PathName, [759](#), [760](#)
  - PathSeparator, [765](#)
  - pathname, [766](#)
  - SameDirectory, [765](#)
  - Split, [766](#)
  - Tail, [766](#)
- FCFSupport::PauseCallback, [775](#)
  - ~PauseCallback, [775](#)
  - Pause, [776](#)
  - PauseCallback, [775](#)
- FCFSupport::PostScriptPrinterDevice, [792](#)
  - ~PostScriptPrinterDevice, [794](#)
  - ClosePrinter, [794](#)
  - lines, [798](#)
  - maxLines, [798](#)
  - needPageHeader, [798](#)
  - NewPage, [794](#)
  - OpenPrinter, [795](#)
  - PSQuote, [795](#)
  - pages, [798](#)
  - partline, [798](#)
  - PostScriptPrinterDevice, [793](#)
  - printerStream, [798](#)
  - Put, [795](#)
  - PutLine, [796](#)
  - PutPageHeader, [796](#)
  - SetTypeSlant, [796](#)
  - SetTypeSpacing, [797](#)
  - SetTypeWeight, [797](#)
  - Tab, [797](#)
  - title, [799](#)
- FCFSupport::PrinterDevice, [801](#)
  - ~PrinterDevice, [804](#)
  - ClosePrinter, [804](#)
  - IsOpenP, [805](#)
  - isOpenP, [809](#)
  - NewPage, [805](#)
  - OpenPrinter, [805](#)
  - PageSize, [802](#)
  - pageSize, [809](#)
  - PrinterDevice, [804](#)
  - PrinterPageSize, [806](#)
  - Put, [806](#), [807](#)
  - PutLine, [807](#)
  - SetTypeSlant, [808](#)
  - SetTypeSpacing, [808](#)
  - SetTypeWeight, [808](#)
  - Tab, [809](#)
  - TypeSlant, [803](#)
  - TypeSpacing, [803](#)
  - TypeWeight, [803](#)
- FCFSupport::ShowBannerCallback, [890](#)
  - ~ShowBannerCallback, [891](#)
  - ShowBanner, [891](#)
  - ShowBannerCallback, [891](#)
- FCFSupport::Station, [941](#)
  - ~Station, [943](#)
  - AppendIndustry, [943](#)
  - Comment, [944](#)
  - comment, [945](#)
  - division, [945](#)
  - industries, [946](#)
  - MyDivision, [944](#)
  - Name, [944](#)
  - name, [946](#)
  - NumberOfIndustries, [944](#)
  - operator=, [944](#)
  - Station, [942](#), [943](#)
  - System, [945](#)
  - TheIndustry, [945](#)
- FCFSupport::SwitchList, [1005](#)
  - ~SwitchList, [1007](#)
  - AddSwitchListElement, [1007](#), [1008](#)
  - DiscardSwitchList, [1008](#)
  - lastIndex, [1011](#)
  - LimitCars, [1008](#)
  - limitCars, [1012](#)
  - NextSwitchListForCarAndIndustry, [1008](#)
  - operator<<, [1011](#)
  - operator[], [1009](#)
  - PickCarEq, [1009](#)
  - PickIndex, [1010](#)

- pickIndex, [1012](#)
- PickLocationEq, [1010](#)
- PickTrainEq, [1010](#)
- ResetLastIndex, [1011](#)
- ResetSwitchList, [1011](#)
- SwitchList, [1007](#)
- theList, [1012](#)
- FCFSupport::SwitchListElement, [1012](#)
  - dropStop, [1018](#)
  - DropStopEQ, [1015](#)
  - DropStopIndustry, [1016](#)
  - DropStopStation, [1016](#)
  - LastTrain, [1016](#)
  - lastTrain, [1018](#)
  - operator=, [1016](#)
  - PickCar, [1017](#)
  - pickCar, [1018](#)
  - pickLoc, [1018](#)
  - PickLocation, [1017](#)
  - PickTrain, [1017](#)
  - pickTrain, [1018](#)
  - SwitchListElement, [1014](#), [1015](#)
  - System, [1017](#)
- FCFSupport::SwitchListElement::StationOrIndustry, [953](#)
  - industry, [953](#)
  - station, [954](#)
- FCFSupport::System, [1023](#)
  - ~System, [1037](#)
  - AddCar, [1037](#)
  - AddOwner, [1038](#)
  - CarAssignment, [1038](#)
  - carDest, [1098](#)
  - carGroups, [1098](#)
  - CarLocationType, [1036](#)
  - CarMovements, [1038](#)
  - carMovements, [1099](#)
  - CarTypeReport, [1036](#)
  - carTypes, [1101](#)
  - CarTypesFile, [1041](#)
  - carTypesFile, [1101](#)
  - CarTypesOrder, [1041](#)
  - carTypesOrder, [1102](#)
  - CarTypesOrderIndex, [1041](#)
  - cars, [1099](#)
  - CarsAtDest, [1039](#)
  - carsAtDest, [1099](#)
  - CarsAtDest\_CarsInTransit, [1039](#)
  - carsAtDest\_carsInTransit, [1099](#)
  - CarsAtWorkBench, [1039](#)
  - carsAtWorkBench, [1099](#)
  - CarsFile, [1039](#)
  - carsFile, [1100](#)
  - CarsInTransit, [1039](#)
  - carsInTransit, [1100](#)
  - CarsMoved, [1040](#)
  - carsMoved, [1100](#)
  - CarsMovedMore, [1040](#)
  - carsMovedMore, [1100](#)
  - CarsMovedOnce, [1040](#)
  - carsMovedOnce, [1100](#)
  - CarsMovedThree, [1040](#)
  - carsMovedThree, [1101](#)
  - CarsMovedTwice, [1040](#)
  - carsMovedTwice, [1101](#)
  - CarsNotMoved, [1041](#)
  - carsNotMoved, [1101](#)
  - curDiv, [1102](#)
  - DeleteAllExistingCars, [1042](#)
  - deliver, [1102](#)
  - divisions, [1102](#)
  - FindCarInCarVector, [1042](#)
  - FindDivisionByIndex, [1042](#)
  - FindDivisionBySymbol, [1043](#)
  - FindDivisionIndex, [1043](#)
  - FindIndustry, [1043](#)
  - FindIndustryByIndex, [1044](#)
  - FindIndustryByName, [1044](#)
  - FindIndustryIndex, [1044](#)
  - FindStationByName, [1045](#)
  - FindStationIndex, [1045](#)
  - FindTrainByIndex, [1045](#)
  - FindTrainByName, [1046](#)
  - FirstCarType, [1046](#)
  - FirstDivision, [1046](#)
  - FirstIndustry, [1046](#)
  - FirstOwner, [1047](#)
  - FirstStation, [1047](#)
  - FirstTrain, [1047](#)
  - FixedRouteMirrorCheck, [1047](#)
  - FormatDutyTime, [1048](#)
  - GetCarStatus, [1048](#)
  - GetIndustryCarCounts, [1048](#)
  - GlobStringMatch, [1048](#)
  - GlobStringMatchHelper, [1049](#)
  - IndRipTrack, [1049](#)
  - IndRipTrackConst, [1049](#)
  - IndScrapYard, [1050](#)
  - indScrapYard, [1102](#)
  - industries, [1103](#)
  - IndustriesFile, [1050](#)
  - industriesFile, [1103](#)
  - IndustryIndex, [1050](#)
  - IndustryTakesCar, [1050](#)
  - InternalRunOneTrain, [1051](#)
  - LastCarType, [1051](#)
  - LastDivision, [1051](#)
  - LastIndustry, [1052](#)
  - LastOwner, [1052](#)

LastStation, 1052  
LastTrain, 1052  
LoadCarFile, 1052  
LoadStatsFile, 1053  
LogCarPickup, 1053  
messageBuffer, 1103  
NextShift, 1053  
numberCars, 1103  
NumberOfCars, 1054  
NumberOfDivisions, 1054  
NumberOfIndustries, 1054  
NumberOfStations, 1054  
NumberOfTrains, 1054  
OrdersFile, 1055  
ordersFile, 1103  
originYard, 1104  
OtherCarOkForTrain, 1055  
owners, 1104  
OwnersFile, 1055  
ownersFile, 1104  
PrintAllCarTypes, 1055  
PrintAllLists, 1056  
PrintAlpha, 1056  
printAlpha, 1104  
PrintAnalysisHeader, 1056  
PrintAtwice, 1057  
printAtwice, 1104  
PrintCarHeading, 1057  
PrintCarTypesHeader, 1057  
PrintCarTypesSummaryHeader, 1057  
PrintDashedLine, 1058  
PrintDispatch, 1058  
printDispatch, 1105  
PrintDispatcher, 1058  
PrintFormFeed, 1059  
PrintIndustryHeader, 1059  
PrintList, 1059  
printList, 1105  
PrintLocCommon, 1060  
PrintLocOneIndustry, 1060  
PrintLtwice, 1060  
printLtwice, 1105  
PrintOneAnalysis, 1061  
PrintOneCarInfo, 1061  
PrintOneCarLocation, 1061  
PrintOneCarType, 1062  
PrintOneIndustry, 1062  
PrintSystemBanner, 1063  
PrintTrainLoc, 1063  
PrintTrainOrderHeader, 1064  
PrintTrainOrders, 1064  
PrintYards, 1064  
printYards, 1105  
Printem, 1059  
printem, 1105  
RanAllTrains, 1064  
ranAllTrains, 1106  
Random, 1065  
Randomize, 1065  
ReLoadCarFile, 1068  
ReadCarTypes, 1065  
ReadDivisions, 1066  
ReadGroupLimit, 1066  
ReadIndustries, 1066  
ReadOwners, 1067  
ReadStations, 1067  
ReadTrainOrders, 1067  
ReadTrains, 1068  
ReportAnalysis, 1068  
ReportCarLocations, 1069  
ReportCarOwners, 1069  
ReportCarTypes, 1070  
ReportCars, 1069  
ReportCarsNotMoved, 1070  
ReportIndustries, 1071  
ReportLocAll, 1071  
ReportLocDivision, 1072  
ReportLocIndustry, 1072  
ReportLocStation, 1073  
ReportTrains, 1073  
ResetIndustryStats, 1073  
RestartLoop, 1074  
RunAllTrains, 1074  
RunBoxMoves, 1074  
RunOneLocal, 1075  
RunOneManifest, 1075  
RunOnePassenger, 1076  
RunOneTrain, 1076  
SaveCars, 1077  
SearchForCarIndexesByNumber, 1077  
SearchForIndustryPattern, 1078  
SearchForTrainPattern, 1078  
SessionNumber, 1078  
sessionNumber, 1106  
SetPrintAlpha, 1078  
SetPrintAtwice, 1080  
SetPrintDispatch, 1080  
SetPrintList, 1081  
SetPrintLtwice, 1081  
SetPrintYards, 1081  
SetPrintem, 1080  
ShiftNumber, 1082  
shiftNumber, 1106  
ShowCarMovements, 1082  
ShowCarsInDivision, 1082  
ShowCarsNotMoved, 1083  
ShowTrainCars, 1083  
ShowTrainTotals, 1084

ShowUnassignedCars, [1084](#)  
 SkipCommentsGets, [1084](#)  
 split, [1085](#)  
 stations, [1106](#)  
 StatsFile, [1085](#)  
 statsFile, [1106](#)  
 StatsPeriod, [1085](#)  
 statsPeriod, [1107](#)  
 StringToInt, [1085](#)  
 StringToIntRange, [1086](#)  
 switchList, [1107](#)  
 System, [1036](#), [1037](#)  
 SystemFile, [1086](#)  
 systemFile, [1107](#)  
 SystemName, [1087](#)  
 systemName, [1107](#)  
 TheCar, [1087](#)  
 TheCarGroup, [1087](#)  
 TheCarType, [1088](#)  
 TheDivision, [1088](#)  
 TheIndustry, [1088](#)  
 TheOwner, [1089](#)  
 TheStation, [1089](#)  
 Today, [1089](#)  
 TotalCars, [1089](#)  
 totalLoads, [1107](#)  
 totalPickups, [1108](#)  
 totalRevenueTons, [1108](#)  
 TotalShifts, [1090](#)  
 totalShifts, [1108](#)  
 totalTons, [1108](#)  
 TrainByIndex, [1090](#)  
 TrainByName, [1090](#)  
 TrainCarPickupCheck, [1091](#)  
 TrainDropAllCars, [1091](#)  
 TrainDropOneCar, [1092](#)  
 trainEmpties, [1108](#)  
 TrainIndex, [1092](#)  
 trainIndex, [1109](#)  
 trainLastLocation, [1109](#)  
 trainLength, [1109](#)  
 trainLoads, [1109](#)  
 TrainLocalDrops, [1093](#)  
 TrainLocalOriginate, [1093](#)  
 TrainLocalPickups, [1094](#)  
 trainLongest, [1109](#)  
 TrainManifestDrops, [1094](#)  
 TrainManifestPickups, [1095](#)  
 TrainPickupOneCar, [1095](#)  
 TrainPrintConsistSummary, [1096](#)  
 TrainPrintFinalSummary, [1096](#)  
 trainPrintOK, [1110](#)  
 TrainPrintTown, [1097](#)  
 trainTons, [1110](#)  
 trains, [1110](#)  
 TrainsFile, [1097](#)  
 trainsFile, [1110](#)  
 trim, [1097](#)  
 UpperCase, [1098](#)  
 wayFreight, [1110](#)  
 whitespace, [1111](#)  
 WriteOneCarToDisk, [1098](#)  
 FCFSupport::TextPrinterDevice, [1111](#)  
   ~TextPrinterDevice, [1112](#)  
   ClosePrinter, [1113](#)  
   currentColumn, [1115](#)  
   NewPage, [1113](#)  
   OpenPrinter, [1113](#)  
   printerStream, [1115](#)  
   Put, [1114](#)  
   PutLine, [1114](#)  
   Tab, [1115](#)  
   TextPrinterDevice, [1112](#)  
 FCFSupport::Train, [1206](#)  
   ~Train, [1210](#)  
   CarTypes, [1211](#)  
   carTypes, [1217](#)  
   Description, [1211](#)  
   description, [1217](#)  
   divList, [1217](#)  
   DivisionList, [1211](#)  
   Done, [1211](#)  
   done, [1217](#)  
   IndustryStop, [1211](#)  
   MaxCars, [1212](#)  
   MaxClear, [1212](#)  
   MaxLength, [1212](#)  
   MaxWeight, [1212](#)  
   maxcars, [1217](#)  
   maxclear, [1218](#)  
   maxlength, [1218](#)  
   maxweight, [1218](#)  
   Name, [1212](#)  
   name, [1218](#)  
   NumberOfOrders, [1213](#)  
   NumberOfStops, [1213](#)  
   OnDuty, [1213](#)  
   onduty, [1218](#)  
   operator=, [1213](#)  
   Order, [1214](#)  
   orders, [1219](#)  
   Print, [1214](#)  
   print, [1219](#)  
   SetMaxLength, [1214](#)  
   SetMaxWeight, [1215](#)  
   SetPrint, [1215](#)  
   SetShift, [1215](#)  
   Shift, [1216](#)



- shift, [1219](#)
  - StationStop, [1216](#)
  - stops, [1219](#)
  - System, [1216](#)
  - Train, [1209](#)
  - TrainType, [1208](#)
  - Type, [1216](#)
  - type, [1219](#)
- FCFSupport::Train::StationOrIndustry, [954](#)
  - industry, [954](#)
  - station, [955](#)
- FCFSupport::TrainDisplayCallback, [1220](#)
  - ~TrainDisplayCallback, [1221](#)
  - CloseTrainDisplay, [1221](#)
  - GrabTrainDisplay, [1221](#)
  - InitializeTrainDisplay, [1221](#)
  - ReleaseTrainDisplay, [1222](#)
  - TrainDisplayCallback, [1220](#)
  - UpdateTrainDisplay, [1222](#)
- FCFSupport::WorkInProgressCallback, [1250](#)
  - ~WorkInProgressCallback, [1250](#)
  - ProgressDone, [1251](#)
  - ProgressStart, [1251](#)
  - ProgressUpdate, [1251](#)
  - WorkInProgressCallback, [1250](#)
- FCFSupportModule, [24](#)
- FEETperMM
  - MRRXtrkCad.tab.h, [1310](#)
- FRAMETYPE\_MASK
  - lcc::MTIHeader, [644](#)
- FRAMETYPE\_SHIFT
  - lcc::MTIHeader, [644](#)
- FUZZ
  - Parsers::TrackGraph::Transform2D, [1226](#)
- fieldflag
  - Parsers::MRRXtrkCad, [633](#)
  - YY\_MRRXtrkCad\_INHERIT, [1279](#)
- fifteenbits
  - lcc, [123](#)
- FileEntry, [93](#)
  - \_destroy, [95](#)
  - \_openFile, [95](#)
  - \_path\_command, [95](#)
  - bind, [96](#)
  - cget, [96](#)
  - configure, [96](#)
  - create, [97](#)
- FileOffset
  - FCFSupport::PDFFileStructures::IndirectObject, [517](#)
- fileOffset
  - FCFSupport::PDFFileStructures::IndirectObject, [520](#)
- Filename
  - TTSupport::TimeTableSystem, [1139](#)
- filepath
  - TTSupport::TimeTableSystem, [1152](#)
- FilterSelectBits
  - CTIAcela, [61](#)
- filterthreshtype
  - CTIAcela, [45](#)
- findAvailableComPorts
  - lcc::CANGridConnectOverUSBSerial, [228](#)
- FindBlock
  - Parsers::TrackGraph, [1181](#)
- FindCab
  - TTSupport::TimeTableSystem, [1139](#)
- FindCarInCarVector
  - FCFSupport::System, [1042](#)
- FindDivisionByIndex
  - FCFSupport::System, [1042](#)
- FindDivisionBySymbol
  - FCFSupport::System, [1043](#)
- FindDivisionIndex
  - FCFSupport::System, [1043](#)
- FindIndustry
  - FCFSupport::System, [1043](#)
- FindIndustryByIndex
  - FCFSupport::System, [1044](#)
- FindIndustryByName
  - FCFSupport::System, [1044](#)
- FindIndustryIndex
  - FCFSupport::System, [1044](#)
- FindNode
  - Parsers::TrackGraph, [1182](#)
- FindOccupied
  - TTSupport::StorageTrack, [987](#)
- FindSegmentIndex
  - Parsers::TrackGraph::CompressedNodeValues, [301](#)
- FindStationByName
  - FCFSupport::System, [1045](#)
  - TTSupport::TimeTableSystem, [1139](#)
- FindStationIndex
  - FCFSupport::System, [1045](#)
- FindStorageTrack
  - TTSupport::Station, [949](#)
  - TTSupport::TimeTableSystem, [1139](#)
- FindTrackTrainsStoredOn
  - TTSupport::Station, [950](#)
- FindTrainByIndex
  - FCFSupport::System, [1045](#)
- FindTrainByName
  - FCFSupport::System, [1046](#)
  - TTSupport::TimeTableSystem, [1140](#)
- FindTrainByNumber
  - TTSupport::TimeTableSystem, [1140](#)
- findtopicintoc
  - HTMLHelp::HTMLHelp, [490](#)
- first
  - lcc, [125](#)

- first\_column
  - yytype, [1283](#)
- first\_line
  - yytype, [1283](#)
- FirstCab
  - TTSupport::TimeTableSystem, [1141](#)
- FirstCarType
  - FCFSupport::System, [1046](#)
- firstChar
  - FCFSupport::PDFFileStructures::Type1Font↔  
Dictionary, [1246](#)
- FirstDivision
  - FCFSupport::System, [1046](#)
- FirstIndustry
  - FCFSupport::System, [1046](#)
- FirstOccupied
  - TTSupport::StorageTrack, [987](#)
- FirstOwner
  - FCFSupport::System, [1047](#)
- FirstPrintOption
  - TTSupport::TimeTableSystem, [1141](#)
- FirstStation
  - FCFSupport::System, [1047](#)
- FirstStorageTrack
  - TTSupport::Station, [950](#)
- FirstTrain
  - FCFSupport::System, [1047](#)
  - TTSupport::TimeTableSystem, [1141](#)
- fivebits
  - lcc, [123](#)
- FixedRouteMirrorCheck
  - FCFSupport::System, [1047](#)
- FixedRouteP
  - FCFSupport::Car, [248](#)
- fixedP
  - FCFSupport::Car, [260](#)
- Flag
  - TTSupport::StationTimes, [957](#)
  - TTSupport::Stop, [979](#)
- flag
  - TTSupport::StationTimes, [958](#)
  - TTSupport::Stop, [983](#)
- FlagType
  - TTSupport::Stop, [977](#)
- font
  - FCFSupport::PDFFileStructures::ResourceDictionary, [849](#)
  - HTMLHelp::HTMLHelp, [490](#)
- fontDescriptor
  - FCFSupport::PDFFileStructures::Type1Font↔  
Dictionary, [1246](#)
- FontDictionary
  - FCFSupport::PDFFileStructures::FontDictionary, [444](#)
- FontFamily
  - OvalWidgets, [139](#)
- Fonts
  - HTMLHelp::HTMLHelp, [510](#)
- ForEveryCab
  - TimeTableSystemTcl, [27](#)
- ForEveryNote
  - TimeTableSystemTcl, [28](#)
- ForEveryPrintOption
  - TimeTableSystemTcl, [28](#)
- ForEveryStation
  - TimeTableSystemTcl, [29](#)
- ForEveryTrain
  - TimeTableSystemTcl, [29](#)
- forget
  - ScrollTabNotebook, [871](#)
- form
  - HTMLHelp::HTMLHelp, [490](#)
- FormatDutyTime
  - FCFSupport::System, [1048](#)
- Forward
  - nce, [134](#)
  - xpressnet, [157](#)
- forward
  - CabWidgets::LocomotiveDirection, [579](#)
  - HTMLHelp::HTMLHelp, [490](#)
- forwardcurrenttopic
  - HTMLHelp::HTMLHelp, [491](#)
- fp
  - Parsers::ParseFile, [752](#)
- fraction
  - OvalWidgets::OvalScrollBar, [727](#)
- FreeObject
  - FCFSupport::PDFFileStructures::CrossReference↔  
Table, [360](#)
- FreedObject
  - FCFSupport::PDFFileStructures::FreedObject, [446](#)
- From
  - TTSupport::Occupied, [680](#)
  - TTSupport::TimeRange, [1126](#)
- from
  - TTSupport::Occupied, [683](#)
  - TTSupport::TimeRange, [1129](#)
- FullPath
  - FCFSupport::PathName, [761](#)
  - TTSupport::PathName, [770](#)
- Function
  - xpressnet::DoubleHeaderInformation, [418](#)
  - xpressnet::LocomotiveInformation, [583](#)
- FunctionStatus
  - xpressnet::FunctionStatus, [449](#)
- FunctionStatusRequest
  - xpressnet::XPressNet, [1264](#)
- fval
  - yy\_MRRXtrkCad\_stype, [1281](#)

- gPos1
  - Parsers::SegVector, [881](#)
- gPos2
  - Parsers::SegVector, [882](#)
- GRSupport, [98](#)
  - \_ROPI, [99](#)
  - \_ROI2, [100](#)
  - DegreesToRadians, [100](#)
  - PI2, [102](#)
  - PI, [101](#)
  - RadiansToDegrees, [100](#)
  - VerifyBooleanMethod, [101](#)
  - VerifyColorMethod, [101](#)
  - VerifyDoubleMethod, [101](#)
  - VerifyIntegerMethod, [101](#)
  - VerifyOrientationHVMMethod, [101](#)
- GRSupportModule, [71](#)
- GScale
  - MRRXtrkCad.tab.h, [1310](#)
- gccomponent
  - lcc::CANGridConnectOverCANSocket, [219](#)
  - lcc::CANGridConnectOverTcp, [223](#)
  - lcc::CANGridConnectOverUSBSerial, [228](#)
- gcmmessage
  - lcc::CANGridConnect, [213](#)
- gcreply
  - lcc::CANGridConnect, [213](#)
- GenerationNumber
  - FCFSupport::PDFFileStructures::IndirectObject, [517](#)
- generationNumber
  - FCFSupport::PDFFileStructures::IndirectObject, [520](#)
- Get
  - linuxgpio::GpioInputActiveLow, [456](#)
  - linuxgpio::GpioOutputSafeHighInvert, [460](#)
  - linuxgpio::GpioOutputSafeLowInverted, [464](#)
  - linuxgpio::LinuxGpio, [572](#)
- get
  - LabelComboBox, [108](#)
  - OvalWidgets::OvalScale, [717](#)
  - OvalWidgets::OvalSlider, [722](#)
  - OvalWidgets::OvalScrollBar, [728](#)
- get3dcolor
  - ScrollTabNotebook, [871](#)
- get\_html
  - HTMLHelp::HTMLHelp, [491](#)
- GetAlert
  - RaildriverIO, [824](#)
- getAliasOfNID
  - lcc::CANGridConnect, [208](#)
- getAllAliases
  - lcc::CANGridConnect, [208](#)
- getAllNIDs
  - lcc::CANGridConnect, [208](#)
- GetAutoBrake
  - RaildriverIO, [824](#)
- GetBailOff
  - RaildriverIO, [824](#)
- GetBell
  - RaildriverIO, [824](#)
- getBits
  - lcc::CANGridConnect, [209](#)
- GetBlueButton1
  - RaildriverIO, [825](#)
- GetBlueButton10
  - RaildriverIO, [825](#)
- GetBlueButton11
  - RaildriverIO, [825](#)
- GetBlueButton12
  - RaildriverIO, [825](#)
- GetBlueButton13
  - RaildriverIO, [825](#)
- GetBlueButton14
  - RaildriverIO, [826](#)
- GetBlueButton15
  - RaildriverIO, [826](#)
- GetBlueButton16
  - RaildriverIO, [826](#)
- GetBlueButton17
  - RaildriverIO, [826](#)
- GetBlueButton18
  - RaildriverIO, [826](#)
- GetBlueButton19
  - RaildriverIO, [827](#)
- GetBlueButton2
  - RaildriverIO, [827](#)
- GetBlueButton20
  - RaildriverIO, [827](#)
- GetBlueButton21
  - RaildriverIO, [827](#)
- GetBlueButton22
  - RaildriverIO, [827](#)
- GetBlueButton23
  - RaildriverIO, [828](#)
- GetBlueButton24
  - RaildriverIO, [828](#)
- GetBlueButton25
  - RaildriverIO, [828](#)
- GetBlueButton26
  - RaildriverIO, [828](#)
- GetBlueButton27
  - RaildriverIO, [828](#)
- GetBlueButton28
  - RaildriverIO, [829](#)
- GetBlueButton3
  - RaildriverIO, [829](#)
- GetBlueButton4
  - RaildriverIO, [829](#)
- GetBlueButton5

- RaildriverIO, [829](#)
- GetBlueButton6
  - RaildriverIO, [829](#)
- GetBlueButton7
  - RaildriverIO, [830](#)
- GetBlueButton8
  - RaildriverIO, [830](#)
- GetBlueButton9
  - RaildriverIO, [830](#)
- getBytes
  - lcc::GridConnectReply, [476](#)
- GetCarStatus
  - FCFSupport::System, [1048](#)
- GetCurveSegment
  - Parsers::BezierBodyElt, [188](#)
  - Parsers::CornuBodyElt, [345](#)
- getData
  - lcc::CanMessage, [236](#)
- GetEBrakeDown
  - RaildriverIO, [830](#)
- GetEBrakeUp
  - RaildriverIO, [830](#)
- getElementsByld
  - SimpleDOMElt, [906](#)
- getElementsByTagName
  - SimpleDOMElt, [906](#)
- getHeader
  - lcc::CANHeader, [231](#)
  - lcc::CanMessage, [236](#)
  - lcc::GridConnectReply, [476](#)
  - lcc::MTIDetail, [637](#)
  - lcc::MTIHeader, [643](#)
- GetHeadlight
  - RaildriverIO, [831](#)
- getHexDigit
  - lcc::GridConnectReply, [476](#)
- GetIndependBrake
  - RaildriverIO, [831](#)
- GetIndustryCarCounts
  - FCFSupport::System, [1048](#)
- GetInstance
  - HTMLHelp::HTMLHelp, [491](#)
- GetKidLimits
  - FCFSupport::PDFFileStructures::PageLabelTree, [744](#)
- GetLI100VersionNumbers
  - xpressnet::XPressNet, [1265](#)
- getMyAlias
  - lcc::CanAlias, [201](#)
- getMyNIDList
  - lcc::CanAlias, [201](#)
- getNIDofAlias
  - lcc::CANGridConnect, [209](#)
- getNextAlias
  - lcc::CanAlias, [202](#)
- GetNextCommandStationResponse
  - xpressnet::XPressNet, [1265](#)
- getNumBytes
  - lcc::GridConnectReply, [477](#)
- GetPanDown
  - RaildriverIO, [831](#)
- GetPanLeft
  - RaildriverIO, [831](#)
- GetPanRight
  - RaildriverIO, [831](#)
- GetPanUp
  - RaildriverIO, [832](#)
- GetPantograph
  - RaildriverIO, [832](#)
- getParent
  - SimpleDOMElt, [906](#)
- GetPrintOption
  - TTSupport::TimeTableSystem, [1141](#)
- GetProductCodeId
  - RaildriverIO, [832](#)
- GetProductId
  - azatrax::Azatrax, [175](#)
- GetProtocolNames
  - lcc::OpenLCBProtocols, [707](#)
- GetRangeDown
  - RaildriverIO, [832](#)
- GetRangeUp
  - RaildriverIO, [832](#)
- GetReverser
  - RaildriverIO, [833](#)
- GetSand
  - RaildriverIO, [833](#)
- GetStateData
  - azatrax::Azatrax, [175](#)
- GetStraightSegment
  - Parsers::BezierBodyElt, [188](#)
  - Parsers::CornuBodyElt, [345](#)
- GetThrottle
  - RaildriverIO, [833](#)
- GetTurnoutCurveSegment
  - Parsers::TurnoutBodyElt, [1233](#)
- GetTurnoutJointSegment
  - Parsers::TurnoutBodyElt, [1234](#)
- GetTurnoutRoute
  - Parsers::TurnoutBodyElt, [1234](#)
- GetTurnoutStraightSegment
  - Parsers::TurnoutBodyElt, [1234](#)
- GetWhistleDown
  - RaildriverIO, [833](#)
- GetWhistleUp
  - RaildriverIO, [833](#)
- GetWiper
  - RaildriverIO, [834](#)

- getZoom
  - CTCPanel::CTCPanel, [382](#)
- GetZoomUp
  - RaildriverIO, [834](#)
- GetZoopDown
  - RaildriverIO, [834](#)
- geti
  - CTCPanel::CTCLabel, [364](#)
  - CTCPanel::CTCPanel, [382](#)
  - CTCPanel::CodeButton, [292](#)
  - CTCPanel::Crossing, [352](#)
  - CTCPanel::Crossover, [356](#)
  - CTCPanel::CurvedBlock, [397](#)
  - CTCPanel::DoubleSlip, [426](#)
  - CTCPanel::EndBumper, [432](#)
  - CTCPanel::HiddenBlock, [481](#)
  - CTCPanel::Lamp, [548](#)
  - CTCPanel::PushButton, [812](#)
  - CTCPanel::SIGPlate, [900](#)
  - CTCPanel::SWPlate, [1021](#)
  - CTCPanel::SchLabel, [857](#)
  - CTCPanel::ScissorCrossover, [861](#)
  - CTCPanel::Signal, [895](#)
  - CTCPanel::SingleSlip, [912](#)
  - CTCPanel::StraightBlock, [995](#)
  - CTCPanel::StubYard, [999](#)
  - CTCPanel::Switch, [1003](#)
  - CTCPanel::ThreeWaySW, [1118](#)
  - CTCPanel::ThroughYard, [1122](#)
  - CTCPanel::Toggle, [1158](#)
- getlistbox
  - LabelComboBox, [108](#)
- gettext, [97](#)
  - \_, [98](#)
  - \_m, [98](#)
  - \_mx, [98](#)
- getv
  - CTCPanel::CTCLabel, [364](#)
  - CTCPanel::CTCPanel, [382](#)
  - CTCPanel::CodeButton, [292](#)
  - CTCPanel::Crossing, [352](#)
  - CTCPanel::Crossover, [356](#)
  - CTCPanel::DoubleSlip, [426](#)
  - CTCPanel::HiddenBlock, [482](#)
  - CTCPanel::Lamp, [548](#)
  - CTCPanel::PushButton, [812](#)
  - CTCPanel::SIGPlate, [900](#)
  - CTCPanel::SWPlate, [1022](#)
  - CTCPanel::SchLabel, [857](#)
  - CTCPanel::ScissorCrossover, [861](#)
  - CTCPanel::Signal, [896](#)
  - CTCPanel::SingleSlip, [912](#)
  - CTCPanel::StraightBlock, [995](#)
  - CTCPanel::StubYard, [999](#)
  - CTCPanel::Switch, [1004](#)
  - CTCPanel::ThreeWaySW, [1118](#)
  - CTCPanel::Toggle, [1158](#)
- getvalue
  - LabelComboBox, [108](#)
  - LabelSpinBox, [118](#)
- GlobStringMatch
  - FCFSupport::System, [1048](#)
- GlobStringMatchHelper
  - FCFSupport::System, [1049](#)
- GpioInputActiveHigh
  - linuxgpio::GpioInputActiveHigh, [453](#)
- GpioInputActiveLow
  - linuxgpio::GpioInputActiveLow, [455](#)
- GpioOutputSafeHigh
  - linuxgpio::GpioOutputSafeHigh, [457](#)
- GpioOutputSafeHighInvert
  - linuxgpio::GpioOutputSafeHighInvert, [459](#)
- GpioOutputSafeLow
  - linuxgpio::GpioOutputSafeLow, [461](#)
- GpioOutputSafeLowInverted
  - linuxgpio::GpioOutputSafeLowInverted, [463](#)
- GrType
  - Parsers::SegVector, [880](#)
- GrabTrainDisplay
  - FCFSupport::TrainDisplayCallback, [1221](#)
- Graph
  - Parsers::TrackGraph, [1174](#)
- Graphics Support Code, [70](#)
- GridConnectMessage
  - lcc::GridConnectMessage, [467](#)
- GridConnectReply
  - lcc::GridConnectReply, [473](#)
- Group
  - FCFSupport::CarGroup, [266](#)
  - FCFSupport::CarType, [270](#)
- group
  - FCFSupport::CarGroup, [267](#)
  - FCFSupport::CarType, [271](#)
- HBar
  - OvalWidgets, [140](#)
- HMalphanumeric
  - HTMLHelp::HTMLHelp, [510](#)
- HMappend\_css
  - HTMLHelp::HTMLHelp, [492](#)
- HMcgiDecode
  - HTMLHelp::HTMLHelp, [492](#)
- HMcgiMap
  - HTMLHelp::HTMLHelp, [492](#)
- HMcheck\_tocRelative
  - HTMLHelp::HTMLHelp, [492](#)
- HMcurrent\_tags
  - HTMLHelp::HTMLHelp, [492](#)

- HMdo\_map
  - [HTMLHelp::HTMLHelp, 493](#)
- HMesc\_map
  - [HTMLHelp::HTMLHelp, 510](#)
- HMevents
  - [HTMLHelp::HTMLHelp, 511](#)
- HMextract\_param
  - [HTMLHelp::HTMLHelp, 493](#)
- HMform\_map
  - [HTMLHelp::HTMLHelp, 511](#)
- HMgot\_image
  - [HTMLHelp::HTMLHelp, 493](#)
- HMgoto
  - [HTMLHelp::HTMLHelp, 493](#)
- HMininit\_state
  - [HTMLHelp::HTMLHelp, 494](#)
- HMininit\_win
  - [HTMLHelp::HTMLHelp, 494](#)
- HMinput\_checkbox
  - [HTMLHelp::HTMLHelp, 494](#)
- HMinput\_hidden
  - [HTMLHelp::HTMLHelp, 494](#)
- HMinput\_image
  - [HTMLHelp::HTMLHelp, 494](#)
- HMinput\_password
  - [HTMLHelp::HTMLHelp, 495](#)
- HMinput\_radio
  - [HTMLHelp::HTMLHelp, 495](#)
- HMinput\_reset
  - [HTMLHelp::HTMLHelp, 495](#)
- HMinput\_submit
  - [HTMLHelp::HTMLHelp, 495](#)
- HMinput\_text
  - [HTMLHelp::HTMLHelp, 495](#)
- HMinsert\_map
  - [HTMLHelp::HTMLHelp, 511](#)
- HMlink\_callback
  - [HTMLHelp::HTMLHelp, 496](#)
- HMlink\_hit
  - [HTMLHelp::HTMLHelp, 496](#)
- HMlink\_setup
  - [HTMLHelp::HTMLHelp, 496](#)
- HMlist\_elements
  - [HTMLHelp::HTMLHelp, 511](#)
- HMload\_css
  - [HTMLHelp::HTMLHelp, 496](#)
- HMmap\_esc
  - [HTMLHelp::HTMLHelp, 497](#)
- HMmap\_reply
  - [HTMLHelp::HTMLHelp, 497](#)
- HMOptimize
  - [HTMLHelp::HTMLHelp, 497](#)
- HMparam\_map
  - [HTMLHelp::HTMLHelp, 511](#)
- HMparse\_html
  - [HTMLHelp::HTMLHelp, 497](#)
- HMrender
  - [HTMLHelp::HTMLHelp, 497](#)
- HMreset\_win
  - [HTMLHelp::HTMLHelp, 498](#)
- HMset\_font
  - [HTMLHelp::HTMLHelp, 498](#)
- HMset\_image
  - [HTMLHelp::HTMLHelp, 498](#)
- HMset\_indent
  - [HTMLHelp::HTMLHelp, 498](#)
- HMset\_state
  - [HTMLHelp::HTMLHelp, 499](#)
- HMstack
  - [HTMLHelp::HTMLHelp, 499](#)
- HMsubmit\_button
  - [HTMLHelp::HTMLHelp, 499](#)
- HMsubmit\_form
  - [HTMLHelp::HTMLHelp, 500](#)
- HMsubmit\_index
  - [HTMLHelp::HTMLHelp, 500](#)
- HMtag\_a
  - [HTMLHelp::HTMLHelp, 500](#)
- HMtag\_color
  - [HTMLHelp::HTMLHelp, 501](#)
- HMtag\_dt
  - [HTMLHelp::HTMLHelp, 501](#)
- HMtag\_font
  - [HTMLHelp::HTMLHelp, 501](#)
- HMtag\_form
  - [HTMLHelp::HTMLHelp, 502](#)
- HMtag\_hmstart
  - [HTMLHelp::HTMLHelp, 502](#)
- HMtag\_hr
  - [HTMLHelp::HTMLHelp, 502](#)
- HMtag\_img
  - [HTMLHelp::HTMLHelp, 502](#)
- HMtag\_input
  - [HTMLHelp::HTMLHelp, 503](#)
- HMtag\_isindex
  - [HTMLHelp::HTMLHelp, 503](#)
- HMtag\_li
  - [HTMLHelp::HTMLHelp, 503](#)
- HMtag\_link
  - [HTMLHelp::HTMLHelp, 504](#)
- HMtag\_map
  - [HTMLHelp::HTMLHelp, 511](#)
- HMtag\_menu
  - [HTMLHelp::HTMLHelp, 504](#)
- HMtag\_ol
  - [HTMLHelp::HTMLHelp, 504](#)
- HMtag\_option
  - [HTMLHelp::HTMLHelp, 504](#)

- HMtag\_select
  - HTMLHelp::HTMLHelp, [504](#)
- HMtag\_textarea
  - HTMLHelp::HTMLHelp, [505](#)
- HMtag\_title
  - HTMLHelp::HTMLHelp, [505](#)
- HMtag\_ul
  - HTMLHelp::HTMLHelp, [505](#)
- HMtest\_parse
  - HTMLHelp::HTMLHelp, [505](#)
- HMwent\_to
  - HTMLHelp::HTMLHelp, [506](#)
- HMwin\_install
  - HTMLHelp::HTMLHelp, [506](#)
- HMx\_font
  - HTMLHelp::HTMLHelp, [506](#)
- HMzap\_white
  - HTMLHelp::HTMLHelp, [506](#)
- HOScale
  - MRRXtrkCad.tab.h, [1310](#)
- HTMLHelp, [102](#)
  - HTMLHelp::HTMLHelp, [488](#)
- HTMLHelp::HTMLHelp, [483](#)
  - \_Close, [489](#)
  - \_SBackward, [489](#)
  - \_SForward, [489](#)
  - \_WidgetMap, [509](#)
  - a, [489](#)
  - back, [489](#)
  - backcurrenttopic, [489](#)
  - color, [490](#)
  - command, [509](#)
  - curtopicindex, [509](#)
  - defaultHelpDirectory, [509](#)
  - defaultHelpWindow, [509](#)
  - defaultTableOfContents, [510](#)
  - findtopicintoc, [490](#)
  - font, [490](#)
  - Fonts, [510](#)
  - form, [490](#)
  - forward, [490](#)
  - forwardcurrenttopic, [491](#)
  - get\_html, [491](#)
  - GetInstance, [491](#)
  - HMalphanumeric, [510](#)
  - HMappend\_css, [492](#)
  - HMcgiDecode, [492](#)
  - HMcgiMap, [492](#)
  - HMcheck\_tocRelative, [492](#)
  - HMcurrent\_tags, [492](#)
  - HMdo\_map, [493](#)
  - HMesc\_map, [510](#)
  - HMevents, [511](#)
  - HMextract\_param, [493](#)
  - HMform\_map, [511](#)
  - HMgot\_image, [493](#)
  - HMgoto, [493](#)
  - HMininit\_state, [494](#)
  - HMininit\_win, [494](#)
  - HMinput\_checkbox, [494](#)
  - HMinput\_hidden, [494](#)
  - HMinput\_image, [494](#)
  - HMinput\_password, [495](#)
  - HMinput\_radio, [495](#)
  - HMinput\_reset, [495](#)
  - HMinput\_submit, [495](#)
  - HMinput\_text, [495](#)
  - HMinsert\_map, [511](#)
  - HMlink\_callback, [496](#)
  - HMlink\_hit, [496](#)
  - HMlink\_setup, [496](#)
  - HMlist\_elements, [511](#)
  - HMload\_css, [496](#)
  - HMmap\_esc, [497](#)
  - HMmap\_reply, [497](#)
  - HMoptimize, [497](#)
  - HMparam\_map, [511](#)
  - HMparse\_html, [497](#)
  - HMrender, [497](#)
  - HMreset\_win, [498](#)
  - HMset\_font, [498](#)
  - HMset\_image, [498](#)
  - HMset\_indent, [498](#)
  - HMset\_state, [499](#)
  - HMstack, [499](#)
  - HMsubmit\_button, [499](#)
  - HMsubmit\_form, [500](#)
  - HMsubmit\_index, [500](#)
  - HMtag\_a, [500](#)
  - HMtag\_color, [501](#)
  - HMtag\_dt, [501](#)
  - HMtag\_font, [501](#)
  - HMtag\_form, [502](#)
  - HMtag\_hmstart, [502](#)
  - HMtag\_hr, [502](#)
  - HMtag\_img, [502](#)
  - HMtag\_input, [503](#)
  - HMtag\_isindex, [503](#)
  - HMtag\_li, [503](#)
  - HMtag\_link, [504](#)
  - HMtag\_map, [511](#)
  - HMtag\_menu, [504](#)
  - HMtag\_ol, [504](#)
  - HMtag\_option, [504](#)
  - HMtag\_select, [504](#)
  - HMtag\_textarea, [505](#)
  - HMtag\_title, [505](#)
  - HMtag\_ul, [505](#)

- HMtest\_parse, [505](#)
- HMwent\_to, [506](#)
- HMwin\_install, [506](#)
- HMx\_font, [506](#)
- HMzap\_white, [506](#)
- HTMLHelp, [488](#)
- help, [491](#)
- helpTopic, [492](#)
- helptext, [510](#)
- helptext\_css, [510](#)
- hmstart, [499](#)
- lastsearch, [511](#)
- menu, [507](#)
- nextlink, [507](#)
- panes, [512](#)
- prevlink, [507](#)
- pushcurrenttopic, [507](#)
- render, [507](#)
- searchbackward, [508](#)
- searchforward, [508](#)
- select, [508](#)
- setDefault, [508](#)
- status, [512](#)
- textscroll, [512](#)
- toc, [512](#)
- toc\_css, [512](#)
- tocscroll, [513](#)
- topicstack, [513](#)
- Url, [513](#)
- handle
  - azatrax::Azatrax, [181](#)
- HardReset
  - nce::NCE, [655](#)
- HardwareVersion
  - xpressnet::LI100VersionNumbers, [567](#)
- HasOffset
  - FCFSupport::PDFFileStructures::IndirectObject, [517](#)
- HasRelays
  - azatrax::MRD, [625](#)
- hashCode
  - lcc::CanMessage, [237](#)
- HaveData
  - CTIAcela, [51](#)
- Hazard
  - FCFSupport::Industry, [530](#)
- hazard
  - FCFSupport::Industry, [535](#)
- header
  - splash, [927](#)
- headerword
  - lcc, [123](#)
- Headlight
  - RaildriverIO, [837](#)
- Heads
  - Parsers::LayoutFile, [558](#)
  - Parsers::TrackGraph, [1182](#)
- heads
  - Parsers::TrackGraph, [1194](#)
- help
  - HTMLHelp::HTMLHelp, [491](#)
- helpTopic
  - HTMLHelp::HTMLHelp, [492](#)
- helptext
  - HTMLHelp::HTMLHelp, [510](#)
- helptext\_css
  - HTMLHelp::HTMLHelp, [510](#)
- HiddenBlock
  - CTCPanel::HiddenBlock, [481](#)
- hide
  - splash, [926](#)
- high
  - linuxgpio, [127](#)
- highbyte
  - CTIAcela, [51](#)
- highest
  - lcc::ConfigOptions, [314](#)
- HighestNode
  - Parsers::LayoutFile, [558](#)
  - Parsers::TrackGraph, [1182](#)
- HighestObjectNumber
  - FCFSupport::PDFFileStructures::CrossReference↔  
Table, [360](#)
- hmstart
  - HTMLHelp::HTMLHelp, [499](#)
- Home
  - FCFSupport::Division, [412](#)
- home
  - FCFSupport::Division, [414](#)
- horizontalScaling
  - FCFSupport::PDFPrinterDevice, [784](#)
- hostLEntry
  - lcc::CANGridConnectOverTcp, [223](#)
  - lcc::OpenLCBOverTcp, [705](#)
- Hours
  - nce, [131](#)
- i
  - TclSocketCANModule, [39](#)
- iElt
  - Parsers::IntegerList, [545](#)
- INCHESperMM
  - MRRXtrkCad.tab.h, [1310](#)
- IScale
  - MRRXtrkCad.tab.h, [1310](#)
- icon
  - splash, [927](#)
- icursor
  - LabelComboBox, [109](#)



- id
  - Parsers::TrackGraph::CompressedNodeValues, [302](#)
  - Parsers::TrackGraph::NodeValues, [676](#)
- idMap
  - Parsers::TrackGraph, [1195](#)
- IdNodeMap
  - Parsers::TrackGraph, [1174](#)
- identify
  - OvalWidgets::OvalScrollBar, [728](#)
- Identify\_1
  - azatrax::Azatrax, [175](#)
- Identify\_1\_2
  - azatrax::MRD, [625](#)
- Identify\_2
  - azatrax::MRD, [625](#)
- IdentifyConsumer
  - lcc::OpenLCBNode, [693](#)
- IdentifyEvents
  - lcc::OpenLCBNode, [693](#)
- IdentifyProducer
  - lcc::OpenLCBNode, [693](#)
- idheaders
  - lcc::ConfigurationEditor, [337](#)
- il
  - yy\_MRRXtrkCad\_stype, [1281](#)
- image
  - splash, [927](#)
- in
  - linuxgpio, [127](#)
- IncludesTime
  - TTSupport::StorageTrack, [987](#)
- IncrementAssignments
  - FCFSupport::Car, [249](#)
- IncrementGenerationNumber
  - FCFSupport::PDFFileStructures::IndirectObject, [518](#)
- IncrementStatsLen
  - FCFSupport::Industry, [530](#)
- IncrementTrips
  - FCFSupport::Car, [249](#)
- IncrmentMovementsThisSession
  - FCFSupport::Car, [249](#)
- IndRipTrack
  - FCFSupport::System, [1049](#)
- IndRipTrackConst
  - FCFSupport::System, [1049](#)
- IndScrapYard
  - FCFSupport::System, [1050](#)
- indScrapYard
  - FCFSupport::System, [1102](#)
- IndependBrake
  - RaildriverIO, [837](#)
- index
  - Parsers::TrackBodyElt, [1166](#)
  - Parsers::TrackGraph::EdgeValues, [429](#)
  - ScrollTabNotebook, [871](#)
- IndirectFloatVector
  - FCFSupport::PDFFileStructures::IndirectFloatVector, [514](#)
- IndirectObject
  - FCFSupport::PDFFileStructures::CrossReference↔Table, [361](#)
  - FCFSupport::PDFFileStructures::IndirectObject, [516](#)
- IndirectObjectDictionary
  - FCFSupport::PDFFileStructures::IndirectObject↔Dictionary, [522](#)
- industries
  - FCFSupport::Station, [946](#)
  - FCFSupport::System, [1103](#)
- IndustriesFile
  - FCFSupport::System, [1050](#)
- industriesFile
  - FCFSupport::System, [1103](#)
- Industry
  - FCFSupport::Industry, [526](#), [528](#)
- industry
  - FCFSupport::SwitchListElement::StationOrIndustry, [953](#)
  - FCFSupport::Train::StationOrIndustry, [954](#)
- IndustryIndex
  - FCFSupport::System, [1050](#)
- IndustryMap
  - FCFSupport, [87](#)
- IndustryStop
  - FCFSupport::Train, [1211](#)
- IndustryTakesCar
  - FCFSupport::System, [1050](#)
- IndustryVector
  - FCFSupport, [87](#)
- info
  - FCFSupport::PDFPrinterDevice, [784](#)
- InformationDirectory
  - FCFSupport::PDFFileStructures::Information↔Directory, [539](#)
- Init
  - cmri::CMri, [282](#)
- InitBoard
  - cmri::CMri, [279](#)
- InitTSegId
  - Parsers::BezierBodyElt, [188](#)
  - Parsers::CornuBodyElt, [345](#)
  - Parsers::TurnoutBodyElt, [1234](#)
- InitializeTrainDisplay
  - FCFSupport::TrainDisplayCallback, [1221](#)
- Initials
  - FCFSupport::Owner, [732](#)
- initials
  - FCFSupport::Owner, [733](#)
- Input\_1\_Enabled

- azatrax::SL2, [916](#)
- azatrax::SR4, [931](#)
- input\_1\_enabled
  - azatrax::SL2::status3\_union, [974](#)
  - azatrax::SR4::status3\_union, [972](#)
- Input\_2\_Enabled
  - azatrax::SL2, [916](#)
  - azatrax::SR4, [932](#)
- input\_2\_enabled
  - azatrax::SL2::status3\_union, [974](#)
  - azatrax::SR4::status3\_union, [972](#)
- Input\_3\_Enabled
  - azatrax::SL2, [916](#)
  - azatrax::SR4, [932](#)
- input\_3\_enabled
  - azatrax::SL2::status3\_union, [974](#)
  - azatrax::SR4::status3\_union, [972](#)
- Input\_4\_Enabled
  - azatrax::SL2, [916](#)
  - azatrax::SR4, [932](#)
- input\_4\_enabled
  - azatrax::SL2::status3\_union, [974](#)
  - azatrax::SR4::status3\_union, [972](#)
- Inputs
  - cmri::CMri, [280](#)
- inputs
  - CmriSupport::CmriNode, [288](#)
- insert
  - ScrollTabNotebook, [872](#)
- InsertBezierTrack
  - Parsers::TrackGraph, [1182](#)
- InsertBlock
  - Parsers::TrackGraph, [1182](#)
- insertCompressedNode
  - Parsers::TrackGraph, [1183](#)
- InsertControl
  - Parsers::TrackGraph, [1183](#)
- InsertCornuTrack
  - Parsers::TrackGraph, [1183](#)
- InsertCurveTrack
  - Parsers::TrackGraph, [1183](#)
- InsertJointTrack
  - Parsers::TrackGraph, [1184](#)
- InsertProtocolBit
  - lcc::OpenLCBProtocols, [708](#)
- InsertSensor
  - Parsers::TrackGraph, [1184](#)
- InsertSignal
  - Parsers::TrackGraph, [1184](#)
- InsertStraightTrack
  - Parsers::TrackGraph, [1184](#)
- InsertSwitchMotor
  - Parsers::TrackGraph, [1185](#)
- InsertTurnOut
  - Parsers::TrackGraph, [1185](#)
- InsertTurnTable
  - Parsers::TrackGraph, [1185](#)
- Instruments, [103](#)
  - CommonOptions, [103](#)
- Instruments::AnalogClock, [166](#)
  - ~AnalogClock, [168](#)
  - AnalogClock, [167](#)
  - settime, [168](#)
- Instruments::CabSignalLamp, [196](#)
  - \_ConfigureFillColor, [198](#)
  - \_ConfigureOutlineColor, [198](#)
  - \_ConfigureSize, [198](#)
  - \_ConfigureXY, [199](#)
  - ~CabSignalLamp, [197](#)
  - CabSignalLamp, [197](#)
  - canvas, [199](#)
  - sx, [199](#)
  - sy, [199](#)
- Instruments::DialInstrument, [398](#)
  - ~DialInstrument, [401](#)
  - dTextX, [401](#)
  - dTextY, [402](#)
  - DialInstrument, [400](#)
  - setvalue, [401](#)
  - ValueRange, [402](#)
- Instruments::DigitalClock, [404](#)
  - ~DigitalClock, [406](#)
  - DigitalClock, [405](#)
  - settime, [406](#)
- Instruments::DigitalInstrument, [406](#)
  - ~DigitalInstrument, [408](#)
  - DigitalInstrument, [408](#)
  - setvalue, [408](#)
- IntAppend
  - Parsers::IntegerList, [544](#)
- IntegerList
  - Parsers::IntegerList, [543](#)
- InternalRunOneTrain
  - FCFSupport::System, [1051](#)
- invalid
  - lcc, [125](#)
- Inverse
  - Parsers::TrackGraph::Transform2D, [1225](#)
- invoke
  - CTCPanel::CTCLabel, [364](#)
  - CTCPanel::CTCPanel, [382](#)
  - CTCPanel::CodeButton, [293](#)
  - CTCPanel::Crossing, [352](#)
  - CTCPanel::Crossover, [356](#)
  - CTCPanel::CurvedBlock, [397](#)
  - CTCPanel::DoubleSlip, [426](#)
  - CTCPanel::EndBumper, [432](#)
  - CTCPanel::HiddenBlock, [482](#)

- CTCPanel::Lamp, [549](#)
- CTCPanel::PushButton, [812](#)
- CTCPanel::SIGPlate, [900](#)
- CTCPanel::SWPlate, [1022](#)
- CTCPanel::SchLabel, [857](#)
- CTCPanel::ScissorCrossover, [861](#)
- CTCPanel::Signal, [896](#)
- CTCPanel::SingleSlip, [912](#)
- CTCPanel::StraightBlock, [995](#)
- CTCPanel::StubYard, [999](#)
- CTCPanel::Switch, [1004](#)
- CTCPanel::ThreeWaySW, [1118](#)
- CTCPanel::ThroughYard, [1122](#)
- CTCPanel::Toggle, [1158](#)
- CabWidgets::LocomotiveDirection, [578](#)
- CabWidgets::LocomotiveSpeed, [591](#)
- CabWidgets::SelectLocomotive, [885](#)
- OvalWidgets::OvalButton, [711](#)
- is\_output
  - linuxgpio::LinuxGpio, [572](#)
- isChild
  - SimpleDOMEElement, [907](#)
- IsCompressed
  - Parsers::LayoutFile, [558](#)
  - Parsers::TrackGraph, [1185](#)
- IsCompressedNode
  - Parsers::LayoutFile, [559](#)
  - Parsers::TrackGraph, [1186](#)
- IsDoneP
  - FCFSupport::Car, [249](#)
- IsEven
  - ReadConfiguration, [146](#)
- IsNodeP
  - Parsers::LayoutFile, [559](#)
  - Parsers::TrackGraph, [1186](#)
- IsNone
  - Parsers::TrackGraph, [1186](#)
- IsOpenP
  - FCFSupport::PrinterDevice, [805](#)
- isOpenP
  - FCFSupport::PrinterDevice, [809](#)
- isRoot
  - FCFSupport::PDFFileStructures::PageLabelTree, [745](#)
- IsThisTheAzatraxWeAreLookingFor
  - azatrax::Azatrax, [176](#)
- itemcget
  - CTCPanel::CTCPanel, [383](#)
- itemconfigure
  - CTCPanel::CTCPanel, [383](#)
- lthStation
  - TTSupport::TimeTableSystem, [1141](#)
- ival
  - yy\_MRRXtrkCad\_stype, [1281](#)
- KamadaKawaiSpringLayoutP
  - Parsers::TrackGraph, [1195](#)
- keywords
  - FCFSupport::PDFFileStructures::Information↔Directory, [541](#)
- kids
  - FCFSupport::PDFFileStructures::PageLabelTree, [745](#)
- L
  - Parsers::SegVector, [882](#)
  - Parsers::TurnoutBodyElt, [1237](#)
- LCARS, [119](#)
- LCCModule, [63](#)
- LCDMessage16
  - nce, [131](#)
- LCDMessage8
  - nce, [131](#)
- LEDCommand
  - RaildriverIO, [837](#)
- LI100Message
  - xpressnet::LI100Message, [565](#)
- LI100VersionNumbers
  - xpressnet::LI100VersionNumbers, [567](#)
- LI101XPressNetAddress
  - xpressnet::LI101XPressNetAddress, [569](#)
- LQ24PrinterDevice
  - FCFSupport::LQ24PrinterDevice, [599](#)
- LabelComboBox, [104](#)
  - \_destroy, [106](#)
  - \_path\_command, [106](#)
  - bind, [106](#)
  - cget, [107](#)
  - configure, [107](#)
  - create, [107](#)
  - get, [108](#)
  - getlistbox, [108](#)
  - getvalue, [108](#)
  - icursor, [109](#)
  - post, [109](#)
  - setvalue, [109](#)
  - unpost, [110](#)
- LabelSelectColor, [110](#)
  - \_destroy, [111](#)
  - \_path\_command, [112](#)
  - cget, [112](#)
  - ColorPopup, [112](#)
  - configure, [114](#)
  - create, [114](#)
- LabelSpinBox, [114](#)
  - \_destroy, [116](#)
  - \_path\_command, [116](#)
  - bind, [116](#)
  - cget, [117](#)

- configure, [117](#)
  - create, [117](#)
  - getvalue, [118](#)
  - setvalue, [118](#)
- labels
  - FCFSupport::PDFFileStructures::CatalogDictionary, [275](#)
- Lamp
  - CTCPanel::Lamp, [547](#)
- LampBits
  - CTIAcela, [62](#)
- lappendCP
  - CTCPanel::CTCPanel, [383](#)
- last
  - lcc, [125](#)
- last\_column
  - yytype, [1283](#)
- last\_line
  - yytype, [1284](#)
- LastCab
  - TTSupport::TimeTableSystem, [1142](#)
- LastCarType
  - FCFSupport::System, [1051](#)
- lastChar
  - FCFSupport::PDFFileStructures::Type1Font↔Dictionary, [1246](#)
- LastDivision
  - FCFSupport::System, [1051](#)
- lastIndex
  - FCFSupport::SwitchList, [1011](#)
- LastIndustry
  - FCFSupport::System, [1052](#)
- lastObjectNumber
  - FCFSupport::PDFFileStructures::CrossReference↔Table, [361](#)
- LastOccupied
  - TTSupport::StorageTrack, [988](#)
- LastOwner
  - FCFSupport::System, [1052](#)
- LastPrintOption
  - TTSupport::TimeTableSystem, [1142](#)
- LastStation
  - FCFSupport::System, [1052](#)
- LastStorageTrack
  - TTSupport::Station, [950](#)
- LastTrain
  - FCFSupport::Car, [249](#)
  - FCFSupport::SwitchListElement, [1016](#)
  - FCFSupport::System, [1052](#)
  - TTSupport::TimeTableSystem, [1142](#)
- lastTrain
  - FCFSupport::SwitchListElement, [1018](#)
- lastsearch
  - HTMLHelp::HTMLHelp, [511](#)
- lasttrain
  - FCFSupport::Car, [260](#)
- Latch\_1
  - azatrax::MRD, [626](#)
- latch\_1
  - azatrax::MRD::status1\_union, [963](#)
- Latch\_2
  - azatrax::MRD, [626](#)
- latch\_2
  - azatrax::MRD::status1\_union, [964](#)
- LayoutFile
  - Parsers::LayoutFile, [553](#)
- Layover
  - TTSupport::Stop, [979](#)
- layover
  - TTSupport::Stop, [984](#)
- lbg
  - ScrollTabNotebook, [876](#)
- lcc, [119](#)
  - AbstractMRMessage, [126](#)
  - AbstractMessage, [125](#)
  - byte, [122](#)
  - bytelist, [122](#)
  - bytelist72, [122](#)
  - complete, [125](#)
  - databuf, [122](#)
  - datagramcontent, [125](#)
  - eightbytes, [123](#)
  - eventvalidity, [125](#)
  - fifteenbits, [123](#)
  - first, [125](#)
  - fivebits, [123](#)
  - headerword, [123](#)
  - invalid, [125](#)
  - last, [125](#)
  - length, [123](#)
  - middle, [125](#)
  - nid, [123](#)
  - sixbits, [124](#)
  - sixteenbits, [124](#)
  - stream, [125](#)
  - threebits, [124](#)
  - twelvebits, [124](#)
  - twobits, [124](#)
  - uint32, [124](#)
  - unknown, [125](#)
  - valid, [125](#)
- lcc::CANGridConnect, [203](#)
  - \_flags0, [206](#)
  - \_messageReader, [206](#)
  - \_reserveMyAlias, [206](#)
  - \_sendDatagram, [207](#)
  - \_sendmessage, [207](#)
  - \_timedout, [207](#)

- [\\_timeout](#), [212](#)
  - [\\_timeoutFlag](#), [213](#)
  - [aliasMap](#), [213](#)
  - [CANGridConnect](#), [205](#)
  - [canheader](#), [213](#)
  - [countNUL](#), [207](#)
  - [datagrambuffers](#), [213](#)
  - [gcmessage](#), [213](#)
  - [gcreply](#), [213](#)
  - [getAliasOfNID](#), [208](#)
  - [getAllAliases](#), [208](#)
  - [getAllNIDs](#), [208](#)
  - [getBits](#), [209](#)
  - [getNIDofAlias](#), [209](#)
  - [listeq](#), [210](#)
  - [messagebuffers](#), [214](#)
  - [messagehandler](#), [214](#)
  - [mtidetail](#), [214](#)
  - [mtiheader](#), [214](#)
  - [mycanalias](#), [214](#)
  - [NIDPATTERN](#), [215](#)
  - [nidMap](#), [214](#)
  - [parent](#), [215](#)
  - [populateAliasMap](#), [210](#)
  - [reserveAlias](#), [211](#)
  - [sendMessage](#), [211](#)
  - [sendOpenLCBMessage](#), [211](#)
  - [setMessageHandler](#), [212](#)
  - [simplenodeflags](#), [215](#)
  - [updateAliasMap](#), [212](#)
- [Icc::CANGridConnectOverCANSocket](#), [215](#)
  - [\\_CancelOpenTransport](#), [217](#)
  - [\\_OpenTransport](#), [217](#)
  - [buildSocketnamenidDialog](#), [218](#)
  - [CANGridConnectOverCANSocket](#), [217](#)
  - [drawOptionsDialog](#), [218](#)
  - [gccomponent](#), [219](#)
  - [nidLEntry](#), [219](#)
  - [requiredOpts](#), [219](#)
  - [socket](#), [219](#)
  - [socketnameLEntry](#), [219](#)
  - [socketnamenidDialog](#), [220](#)
- [Icc::CANGridConnectOverTcp](#), [220](#)
  - [\\_CancelOpenTransport](#), [222](#)
  - [\\_OpenTransport](#), [222](#)
  - [buildPortnidandhostDialog](#), [222](#)
  - [CANGridConnectOverTcp](#), [221](#)
  - [drawOptionsDialog](#), [222](#)
  - [gccomponent](#), [223](#)
  - [hostLEntry](#), [223](#)
  - [nidLEntry](#), [224](#)
  - [portLSpin](#), [224](#)
  - [portnidandhostDialog](#), [224](#)
  - [requiredOpts](#), [223](#)
  - [socket](#), [224](#)
- [Icc::CANGridConnectOverUSBSerial](#), [224](#)
  - [\\_CancelOpenTransport](#), [226](#)
  - [\\_OpenTransport](#), [226](#)
  - [buildPortandnidDialog](#), [227](#)
  - [CANGridConnectOverUSBSerial](#), [226](#)
  - [drawOptionsDialog](#), [227](#)
  - [findAvailableComPorts](#), [228](#)
  - [gccomponent](#), [228](#)
  - [nidLEntry](#), [228](#)
  - [portLCombo](#), [229](#)
  - [portandnidDialog](#), [229](#)
  - [requiredOpts](#), [228](#)
  - [ttyfd](#), [229](#)
- [Icc::CANHeader](#), [229](#)
  - [CANHeader](#), [230](#)
  - [getHeader](#), [231](#)
  - [OPENLCBFRAME\\_MASK](#), [231](#)
  - [OPENLCBFRAME\\_SHIFT](#), [231](#)
  - [RESERVED\\_SHIFT](#), [232](#)
  - [SRCID\\_MASK](#), [232](#)
  - [SRCID\\_SHIFT](#), [232](#)
  - [setHeader](#), [231](#)
  - [VARIABLEFIELD\\_MASK](#), [232](#)
  - [VARIABLEFIELD\\_SHIFT](#), [232](#)
- [Icc::CanAlias](#), [200](#)
  - [\\_peelnid](#), [201](#)
  - [CanAlias](#), [201](#)
  - [getMyAlias](#), [201](#)
  - [getMyNIDList](#), [201](#)
  - [getNextAlias](#), [202](#)
  - [lfsr1](#), [202](#)
  - [lfsr2](#), [202](#)
  - [myalias](#), [203](#)
  - [nidlist](#), [203](#)
  - [validate](#), [202](#)
- [Icc::CanMessage](#), [233](#)
  - [\\_header](#), [239](#)
  - [CanMessage](#), [235](#)
  - [copy](#), [236](#)
  - [equals](#), [236](#)
  - [getData](#), [236](#)
  - [getHeader](#), [236](#)
  - [hashCode](#), [237](#)
  - [replyExpected](#), [237](#)
  - [setData](#), [237](#)
  - [setHeader](#), [238](#)
  - [setNumDataElements](#), [238](#)
  - [toString](#), [238](#)
  - [validate](#), [238](#)
- [Icc::CanTransport](#), [239](#)
  - [\\_transportlayerconf](#), [240](#)
  - [CanTransport](#), [240](#)
  - [transport](#), [241](#)

- lcc::ConfigMemory, 303
  - \_Close, 305
  - \_Dump, 306
  - \_Read, 307
  - \_Restore, 308
  - \_Write, 308
  - \_datagramhandler, 305
  - \_datagramrejecterror, 309
  - \_dumpAsHex, 306
  - \_dumpAsText, 306
  - \_getAddressRange, 307
  - \_ioComplete, 309
  - \_messagehandler, 307
  - \_readmemory, 307
  - \_spaces, 309
  - \_writememory, 308
  - address, 310
  - ConfigMemory, 305
  - count, 310
  - datagrambuffer, 310
  - olddatagramhandler, 310
  - oldgeneralmessagehandler, 310
  - putdebug, 309
  - readlist, 310
  - space, 311
  - writeReplyCheck, 311
  - writelist, 311
- lcc::ConfigOptions, 311
  - \_Close, 313
  - available, 314
  - ConfigOptions, 312
  - highest, 314
  - lowest, 314
  - name, 314
  - nodeid, 314
  - putdebug, 313
  - writelengths, 314
- lcc::ConfigurationEditor, 315
  - \_close, 319
  - \_datagramhandler, 319
  - \_datagramrejecterror, 335
  - \_eventidComboRead, 320
  - \_eventidComboWrite, 320
  - \_eventidEntryRead, 320
  - \_eventidEntryWrite, 321
  - \_eventidnumber, 335
  - \_groupnumber, 335
  - \_intComboRead, 321
  - \_intComboWrite, 322
  - \_intSpinRead, 322
  - \_intSpinWrite, 323
  - \_intnumber, 335
  - \_ioComplete, 336
  - \_menu, 336
  - \_mkbuttons, 336
  - \_printexport, 323
  - \_printexport\_csv, 323
  - \_printexport\_csv\_frame, 324
  - \_printexport\_csv\_frameAcross, 324
  - \_printexport\_csv\_framesAcross, 325
  - \_printexport\_csv\_vframe, 325
  - \_printexport\_csv\_vframeAcross, 325
  - \_printexport\_pdf, 326
  - \_printexport\_pdf\_frame, 326
  - \_printexport\_pdf\_newpage, 327
  - \_printexport\_pdf\_vframe, 327
  - \_printexport\_txt, 328
  - \_printexport\_txt\_frame, 328
  - \_printexport\_txt\_vframe, 329
  - \_printexport\_xml, 329
  - \_printexport\_xml\_frame, 329
  - \_printexport\_xml\_vframe, 331
  - \_processXMLnode, 331
  - \_readall, 332, 336
  - \_readmemory, 332
  - \_segmentnumber, 336
  - \_stringComboRead, 332
  - \_stringComboWrite, 333
  - \_stringEntryRead, 333
  - \_stringEntryWrite, 334
  - \_stringnumber, 336
  - \_writememory, 334
  - buttons, 337
  - cdi, 337
  - ConfigurationEditor, 318
  - datagrambuffer, 337
  - editframe, 337
  - idheaders, 337
  - main, 337
  - olddatagramhandler, 338
  - printexportfiletypes, 338
  - putdebug, 335
  - scroll, 338
  - statusline, 338
  - writeReplyCheck, 338
- lcc::EventID\_or\_null, 437
  - validate, 437
- lcc::EventID, 434
  - \_eventID, 437
  - \_getEventID, 435
  - \_setEventID, 436
  - EVENTIDFMT, 437
  - EventID, 435
  - validate, 436
- lcc::EventLog, 438
  - \_clear, 439
  - \_close, 439
  - \_sendtheevent, 439

- EventLog, [439](#)
- eventReceived, [440](#)
- logscroll, [440](#)
- logtext, [440](#)
- open, [440](#)
- sendevent, [440](#)
- Icc::EventReceived, [441](#)
  - \_Close, [442](#)
  - EventReceived, [441](#)
  - eventid, [442](#)
- Icc::GridConnectMessage, [465](#)
  - \_copyCM, [468](#)
  - \_get\_extended, [468](#)
  - \_get\_rtr, [469](#)
  - \_set\_extended, [469](#)
  - \_set\_rtr, [469](#)
  - GridConnectMessage, [467](#)
  - setByte, [470](#)
  - setHeader, [470](#)
  - setHexDigit, [470](#)
- Icc::GridConnectReply, [471](#)
  - \_RTRoffset, [478](#)
  - \_copyGCM, [474](#)
  - \_get\_extended, [474](#)
  - \_get\_rtr, [475](#)
  - basicFormatCheck, [475](#)
  - createReply, [475](#)
  - getByte, [476](#)
  - getHeader, [476](#)
  - getHexDigit, [476](#)
  - getNumBytes, [477](#)
  - GridConnectReply, [473](#)
  - MAXLEN, [478](#)
  - maxSize, [477](#)
  - setData, [477](#)
  - setElement, [478](#)
- Icc::MTIDetail, [635](#)
  - ADDRESSP\_MASK, [638](#)
  - ADDRESSP\_SHIFT, [638](#)
  - DESTID\_MASK, [638](#)
  - DESTID\_SHIFT, [639](#)
  - EVENTP\_MASK, [639](#)
  - EVENTP\_SHIFT, [639](#)
  - getHeader, [637](#)
  - MODIFIER\_MASK, [639](#)
  - MODIFIER\_SHIFT, [639](#)
  - MTIDetail, [636](#)
  - mtiheader, [639](#)
  - PRIORITY\_MASK, [640](#)
  - PRIORITY\_SHIFT, [640](#)
  - SIMPLE\_MASK, [640](#)
  - SIMPLE\_SHIFT, [640](#)
  - SPECIAL\_MASK, [640](#)
  - STREAMDG\_MASK, [640](#)
  - setHeader, [638](#)
  - TYPEWITHIN\_MASK, [641](#)
  - TYPEWITHIN\_SHIFT, [641](#)
- Icc::MTIHeader, [641](#)
  - canheader, [644](#)
  - FRAMETYPE\_MASK, [644](#)
  - FRAMETYPE\_SHIFT, [644](#)
  - getHeader, [643](#)
  - MTI\_CAN\_MASK, [644](#)
  - MTI\_CAN\_SHIFT, [644](#)
  - MTIHeader, [642](#)
  - setHeader, [643](#)
- Icc::OpenLCBMessage, [684](#)
  - \_cgetdata, [685](#)
  - \_configuredata, [686](#)
  - OpenLCBMessage, [685](#)
  - toString, [686](#)
  - validate, [686](#)
- Icc::OpenLCBNode, [687](#)
  - \_CancelSelectTransport, [690](#)
  - \_SelectTransport, [691](#)
  - \_buildSelectTransportConstructorDialog, [690](#)
  - \_messageHandler, [691](#)
  - \_transportConstructors, [699](#)
  - constructorCombo, [699](#)
  - ConsumerIdentified, [691](#)
  - ConsumerRangeIdentified, [692](#)
  - DatagramReceivedOK, [692](#)
  - DatagramRejected, [692](#)
  - IdentifyConsumer, [693](#)
  - IdentifyEvents, [693](#)
  - IdentifyProducer, [693](#)
  - LearnEvent, [694](#)
  - nidlist, [694](#)
  - OpenLCBNode, [689](#)
  - ProduceEvent, [694](#)
  - ProducerIdentified, [694](#)
  - ProducerRangeIdentified, [695](#)
  - ProtocolSupportRequest, [695](#)
  - protocolsupport, [699](#)
  - ReturnMySimpleNodeInfo, [695](#)
  - ReturnMySupportedProtocols, [696](#)
  - selectTransportConstructor, [696](#)
  - selectTransportConstructorDialog, [699](#)
  - SendDatagram, [696](#)
  - SendInitComplete, [697](#)
  - SendMyNodeVerification, [697](#)
  - SendMySimpleNodeInfo, [697](#)
  - SendMySupportedProtocols, [697](#)
  - SendSimpleNodeInfoRequest, [698](#)
  - SendSupportedProtocolsRequest, [698](#)
  - SendVerifyNodeID, [698](#)
  - simplenodeinfo, [700](#)
  - transport, [700](#)

- transportConstructors, 699
- lcc::OpenLCBOverTcp, 700
  - \_CancelOpenTransport, 702
  - \_OpenTransport, 702
  - \_timeout, 705
- buildPortnidandhostDialog, 703
- datagrambuffers, 705
- drawOptionsDialog, 703
- hostLEntry, 705
- messagebuffers, 705
- messagehandler, 705
- mtidetail, 706
- NIDPATTERN, 706
- nidLEntry, 706
- OpenLCBOverTcp, 702
- portLSpin, 706
- portnidandhostDialog, 706
- requiredOpts, 704
- sendMessage, 704
- setMessageHandler, 704
- sock, 706
- lcc::OpenLCBProtocols, 707
  - bitstype, 709
  - GetProtocolNames, 707
  - InsertProtocolBit, 708
  - ProtocolLabelString, 708
  - protocolstrings, 709
  - validate, 708
- lcc::SendEvent, 886
  - \_Close, 887
  - \_Send, 887
  - eventid, 888
  - SendEvent, 887
- lcc::nid\_or\_null, 672
  - validate, 673
- LdLmt
  - FCFSupport::Car, 250
- ldlmt
  - FCFSupport::Car, 260
- LearnEvent
  - lcc::OpenLCBNode, 694
- leds
  - raildriver::RaildriverClient, 815
- left
  - ScrollTabNotebook, 877
- leftbuttons
  - CabWidgets::LocomotiveSpeed, 593
- len0
  - BezierBody.h, 1301
  - CornuBody.h, 1302
  - Parsers::SegVector, 882
  - TurnoutBody.h, 1314
- len1
  - BezierBody.h, 1301
  - CornuBody.h, 1302
  - Parsers::SegVector, 882
  - TurnoutBody.h, 1314
- Length
  - FCFSupport::Car, 250
- length
  - FCFSupport::Car, 260
  - lcc, 123
  - Parsers::SegVector, 882
  - Parsers::TrackGraph::CompressedEdgeValues, 300
  - Parsers::TrackGraph::EdgeValues, 429
  - Parsers::TrackGraph::NodeValues, 676
  - SimpleDOMEElement, 907
- LengthOfCurve
  - Parsers::TrackGraph, 1186
- LengthOfJoint
  - Parsers::TrackGraph, 1187
- LengthOfNode
  - Parsers::LayoutFile, 559
  - Parsers::TrackGraph, 1187
- LengthOfStraight
  - Parsers::TrackGraph, 1187
- lever
  - CTCPanel::Toggle, 1160
- leverMethods
  - CTCPanel, 81
- If
  - CabWidgets::SelectLocomotive, 886
- lfsr1
  - lcc::CanAlias, 202
- lfsr2
  - lcc::CanAlias, 202
- LimitCars
  - FCFSupport::SwitchList, 1008
- limitCars
  - FCFSupport::SwitchList, 1012
- line\_buffer
  - Parsers::ParseFile, 752
- lines
  - FCFSupport::PDFPrinterDevice, 785
  - FCFSupport::PostScriptPrinterDevice, 798
- LinuxGpio, 67
  - linuxgpio::LinuxGpio, 571
- linuxgpio, 126
  - high, 127
  - in, 127
  - low, 127
  - out, 127
  - pindirection, 127
  - pinnotype, 127
- linuxgpio::GpioInputActiveHigh, 452
  - ~GpioInputActiveHigh, 453
  - basepin, 454
  - GpioInputActiveHigh, 453



- linuxgpio::GpioInputActiveLow, [454](#)
  - ~GpioInputActiveLow, [455](#)
  - basepin, [456](#)
  - Get, [456](#)
  - GpioInputActiveLow, [455](#)
- linuxgpio::GpioOutputSafeHigh, [456](#)
  - ~GpioOutputSafeHigh, [457](#)
  - basepin, [458](#)
  - GpioOutputSafeHigh, [457](#)
- linuxgpio::GpioOutputSafeHighInvert, [458](#)
  - ~GpioOutputSafeHighInvert, [459](#)
  - basepin, [460](#)
  - Clr, [460](#)
  - Get, [460](#)
  - GpioOutputSafeHighInvert, [459](#)
  - Set, [460](#)
- linuxgpio::GpioOutputSafeLow, [461](#)
  - ~GpioOutputSafeLow, [462](#)
  - basepin, [462](#)
  - GpioOutputSafeLow, [461](#)
- linuxgpio::GpioOutputSafeLowInverted, [463](#)
  - ~GpioOutputSafeLowInverted, [464](#)
  - basepin, [465](#)
  - Clr, [464](#)
  - Get, [464](#)
  - GpioOutputSafeLowInverted, [463](#)
  - Set, [465](#)
- linuxgpio::LinuxGpio, [570](#)
  - ~LinuxGpio, [571](#)
  - Clr, [571](#)
  - DIRECTIONFMT, [573](#)
  - EXPORT, [573](#)
  - Get, [572](#)
  - is\_output, [572](#)
  - LinuxGpio, [571](#)
  - read, [572](#)
  - Set, [572](#)
  - UNEXPORT, [573](#)
  - VALUEFMT, [573](#)
  - write, [572](#)
- listeq
  - Icc::CANGridConnect, [210](#)
- Load
  - FCFSupport::Car, [250](#)
- LoadCarFile
  - FCFSupport::System, [1052](#)
- LoadStatsFile
  - FCFSupport::System, [1053](#)
- loadTypes
  - FCFSupport::Industry, [535](#)
- LoadedP
  - FCFSupport::Car, [250](#)
- loadedP
  - FCFSupport::Car, [260](#)
- LoadsAccepted
  - FCFSupport::Industry, [530](#)
- localtime\_r
  - PDFPrinterSupport.h, [1295](#)
- Location
  - FCFSupport::Car, [250](#)
- location
  - FCFSupport::Car, [261](#)
- LocoAddress
  - nce, [131](#)
- locoList
  - CabWidgets::SelectLocomotive, [886](#)
- LocomotiveAddress
  - xpressnet::LocomotiveAddress, [574](#)
- LocomotiveDirection
  - CabWidgets::LocomotiveDirection, [577](#)
- LocomotiveInformation
  - xpressnet::LocomotiveInformation, [581](#)
- LocomotiveInformationRequest
  - xpressnet::XPressNet, [1265](#)
- LocomotiveSpeed
  - CabWidgets::LocomotiveSpeed, [589](#)
- LogCarPickup
  - FCFSupport::System, [1053](#)
- LogMessage
  - FCFSupport::LogMessageCallback, [596](#)
- LogMessageCallback
  - FCFSupport::LogMessageCallback, [596](#)
- logscroll
  - Icc::EventLog, [440](#)
- logtext
  - Icc::EventLog, [440](#)
- lookup\_word
  - Parsers::MRRXtrkCad, [632](#)
  - YY\_MRRXtrkCad\_INHERIT, [1278](#)
- low
  - linuxgpio, [127](#)
- lowbyte
  - CTIAcela, [52](#)
- Lower
  - xpressnet, [158](#)
- lowest
  - Icc::ConfigOptions, [314](#)
- LowestNode
  - Parsers::LayoutFile, [559](#)
  - Parsers::TrackGraph, [1187](#)
- lp
  - Parsers::ParseFile, [753](#)
- IremoveCP
  - CTCPanel::CTCPanel, [384](#)
- LtWt
  - FCFSupport::Car, [251](#)
- ltwt
  - FCFSupport::Car, [261](#)

- MAXLEN
  - lcc::GridConnectReply, [478](#)
- METERSperMM
  - MRRXtrkCad.tab.h, [1310](#)
- MODIFIER\_MASK
  - lcc::MTIDetail, [639](#)
- MODIFIER\_SHIFT
  - lcc::MTIDetail, [639](#)
- MRRXtrkCad
  - Parsers::MRRXtrkCad, [632](#)
- MRRXtrkCad.tab.h
  - BISON\_YYLTYPE\_ISDECLARED, [1305](#)
  - CENTIMETERSperMM, [1309](#)
  - FALSE, [1305](#)
  - FEETperMM, [1310](#)
  - GScale, [1310](#)
  - HOScale, [1310](#)
  - INCHESperMM, [1310](#)
  - IScale, [1310](#)
  - METERSperMM, [1310](#)
  - NScale, [1310](#)
  - OScale, [1311](#)
  - RADIANS, [1305](#)
  - TRUE, [1305](#)
  - YARDSperMM, [1311](#)
  - YY\_MRRXtrkCad\_CHAR, [1305](#)
  - YY\_MRRXtrkCad\_CLASS, [1305](#)
  - YY\_MRRXtrkCad\_COMPATIBILITY, [1305](#)
  - YY\_MRRXtrkCad\_CONSTRUCTOR\_CODE, [1306](#)
  - YY\_MRRXtrkCad\_CONSTRUCTOR\_INIT, [1306](#)
  - YY\_MRRXtrkCad\_CONSTRUCTOR\_PARAM, [1306](#)
  - YY\_MRRXtrkCad\_DEBUG\_FLAG, [1306](#)
  - YY\_MRRXtrkCad\_DEBUG, [1306](#)
  - YY\_MRRXtrkCad\_ERROR\_BODY, [1307](#)
  - YY\_MRRXtrkCad\_ERROR\_VERBOSE, [1307](#)
  - YY\_MRRXtrkCad\_ERROR, [1306](#)
  - YY\_MRRXtrkCad\_INHERIT, [1307](#)
  - YY\_MRRXtrkCad\_LEX\_BODY, [1307](#)
  - YY\_MRRXtrkCad\_LEX, [1307](#)
  - YY\_MRRXtrkCad\_LLOC, [1307](#)
  - YY\_MRRXtrkCad\_LSP\_NEEDED, [1307](#)
  - YY\_MRRXtrkCad\_LTYPE, [1308](#)
  - YY\_MRRXtrkCad\_LVAL, [1308](#)
  - YY\_MRRXtrkCad\_MEMBERS, [1308](#)
  - YY\_MRRXtrkCad\_NERRS, [1308](#)
  - YY\_MRRXtrkCad\_PARSE\_PARAM, [1308](#)
  - YY\_MRRXtrkCad\_PARSE, [1308](#)
  - YY\_MRRXtrkCad\_STYPE, [1309](#)
  - YY\_MRRXtrkCad\_USE\_CONST\_TOKEN, [1309](#)
  - YY\_MRRXtrkCad\_USE\_GOTO, [1309](#)
  - YY\_USE\_CLASS, [1309](#)
  - yytype, [1309](#)
- MRD
  - azatrax::Azatrax, [180](#)
  - azatrax::MRD, [623](#)
- MTI\_CAN\_MASK
  - lcc::MTIHeader, [644](#)
- MTI\_CAN\_SHIFT
  - lcc::MTIHeader, [644](#)
- MTIDetail
  - lcc::MTIDetail, [636](#)
- MTIHeader
  - lcc::MTIHeader, [642](#)
- MTR
  - xpressnet::LocomotiveInformation, [583](#)
- MacroCommand
  - nce::NCE, [655](#)
- MacroNumber
  - nce, [131](#)
- main
  - lcc::ConfigurationEditor, [337](#)
- mainwindow, [604](#)
  - buttons, [619](#)
  - buttons\_add, [609](#)
  - buttons\_delete, [609](#)
  - buttons\_hide, [609](#)
  - buttons\_insert, [609](#)
  - buttons\_itemconfigure, [610](#)
  - buttons\_show, [610](#)
  - mainwindow, [608](#)
  - menu\_activate, [610](#)
  - menu\_add, [611](#)
  - menu\_delete, [611](#)
  - menu\_entrycget, [611](#)
  - menu\_entryconfigure, [612](#)
  - menu\_index, [612](#)
  - menu\_insert, [612](#)
  - menu\_invoke, [613](#)
  - menu\_sethelpvar, [613](#)
  - menu\_type, [613](#)
  - numtoolbars, [619](#)
  - panewindow, [619](#)
  - progress, [619](#)
  - scrollwindow, [620](#)
  - setprogress, [614](#)
  - setstatus, [614](#)
  - showit, [614](#)
  - slideout\_add, [615](#)
  - slideout\_getframe, [615](#)
  - slideout\_hide, [615](#)
  - slideout\_isshownp, [616](#)
  - slideout\_list, [616](#)
  - slideout\_reqwidth, [616](#)
  - slideout\_show, [616](#)
  - slideouts, [620](#)
  - status, [620](#)
  - toolbar\_add, [617](#)
  - toolbar\_addbutton, [617](#)

- toolbar\_buttoncget, [617](#)
- toolbar\_buttonconfigure, [618](#)
- toolbar\_hide, [618](#)
- toolbar\_setbuttonstate, [618](#)
- toolbar\_show, [619](#)
- toolbars, [620](#)
- wipmessage, [620](#)
- Major
  - xpressnet::SoftwareVersion, [922](#)
- MakeCurveSegment
  - Parsers::BezierBodyElt, [189](#)
  - Parsers::CornuBodyElt, [346](#)
- MakeStraightSegment
  - Parsers::BezierBodyElt, [189](#)
  - Parsers::CornuBodyElt, [346](#)
- MakeTimeTableGroupByClass
  - TTSupport::TimeTableSystem, [1142](#)
- MakeTimeTableGroupManually
  - TTSupport::TimeTableSystem, [1144](#)
- MakeTimeTableOneTable
  - TTSupport::TimeTableSystem, [1144](#)
- MakeTimeTableOneTableStationsCenter
  - TTSupport::TimeTableSystem, [1146](#)
- MakeTimeTableOneTableStationsLeft
  - TTSupport::TimeTableSystem, [1146](#)
- MakeTrackEnd
  - Parsers::BezierBodyElt, [189](#)
  - Parsers::CornuBodyElt, [346](#)
- MakeTurnoutCurveSegment
  - Parsers::TurnoutBodyElt, [1235](#)
- MakeTurnoutEnd
  - Parsers::TurnoutBodyElt, [1235](#)
- MakeTurnoutGraphic
  - Parsers::TrackGraph, [1188](#)
- MakeTurnoutJointSegment
  - Parsers::TurnoutBodyElt, [1235](#)
- MakeTurnoutRoute
  - Parsers::TurnoutBodyElt, [1235](#)
- MakeTurnoutRouteList
  - Parsers::TrackGraph, [1188](#)
- MakeTurnoutStraightSegment
  - Parsers::TurnoutBodyElt, [1236](#)
- Manual
  - xpressnet, [158](#)
- Marks
  - FCFSupport::Car, [251](#)
- marks
  - FCFSupport::Car, [261](#)
- mask
  - raildriver::RaildriverClient, [816](#)
- matrix
  - Parsers::TrackGraph::Transform2D, [1227](#)
- MaxCarLen
  - FCFSupport::Industry, [530](#)
- maxCarLen
  - FCFSupport::Industry, [535](#)
- MaxCars
  - FCFSupport::Train, [1212](#)
- MaxClear
  - FCFSupport::Train, [1212](#)
- MaxLength
  - FCFSupport::Train, [1212](#)
- maxLines
  - FCFSupport::PDFPrinterDevice, [785](#)
  - FCFSupport::PostScriptPrinterDevice, [798](#)
- MaxPlate
  - FCFSupport::Industry, [530](#)
- maxSize
  - lcc::GridConnectReply, [477](#)
- MaxWeight
  - FCFSupport::Train, [1212](#)
- MaxWeightClass
  - FCFSupport::Industry, [531](#)
- maxcars
  - FCFSupport::Train, [1217](#)
- maxclear
  - FCFSupport::Train, [1218](#)
- maxlength
  - FCFSupport::Train, [1218](#)
- maxtries
  - CTIAcela, [62](#)
- maxweight
  - FCFSupport::Train, [1218](#)
- maxX
  - Parsers::TurnoutGraphic, [1240](#)
- maxY
  - Parsers::TurnoutGraphic, [1240](#)
- mediaBox
  - FCFSupport::PDFFileStructures::Page, [737](#)
  - FCFSupport::PDFFileStructures::PageTree, [748](#)
- menu
  - HTMLHelp::HTMLHelp, [507](#)
- menu\_activate
  - mainwindow, [610](#)
- menu\_add
  - mainwindow, [611](#)
- menu\_delete
  - mainwindow, [611](#)
- menu\_entrycget
  - mainwindow, [611](#)
- menu\_entryconfigure
  - mainwindow, [612](#)
- menu\_index
  - mainwindow, [612](#)
- menu\_insert
  - mainwindow, [612](#)
- menu\_invoke
  - mainwindow, [613](#)

- menu\_sethelpvar
  - mainwindow, [613](#)
- menu\_type
  - mainwindow, [613](#)
- messageBuffer
  - FCFSupport::System, [1103](#)
- MessageType
  - FCFSupport::LogMessageCallback, [595](#)
  - xpressnet::LI100Message, [566](#)
- MessageTypeCode
  - xpressnet, [157](#)
- messagebuffers
  - lcc::CANGridConnect, [214](#)
  - lcc::OpenLCBOverTcp, [705](#)
- messagehandler
  - lcc::CANGridConnect, [214](#)
  - lcc::OpenLCBOverTcp, [705](#)
- middle
  - lcc, [125](#)
- Minor
  - Parsers::TrackGraph::Transform2D, [1225](#)
  - xpressnet::SoftwareVersion, [922](#)
- Minutes
  - nce, [132](#)
- minX
  - Parsers::TurnoutGraphic, [1240](#)
- minY
  - Parsers::TurnoutGraphic, [1241](#)
- mirror
  - FCFSupport::Industry, [535](#)
- mirrorP
  - FCFSupport::Car, [261](#)
- modificationDate
  - FCFSupport::PDFFileStructures::Information←  
Directory, [541](#)
- modtype
  - azatrax::MRD::status1\_union, [964](#)
- MomentumLevel
  - nce, [132](#)
- momtype
  - CTIAcela, [45](#)
- Motor\_1\_Direction
  - azatrax::SL2, [917](#)
- Motor\_1\_State
  - azatrax::SL2, [917](#)
- motor\_1\_direction
  - azatrax::SL2::status1\_union, [961](#)
- motor\_1\_state
  - azatrax::SL2::status1\_union, [961](#)
- Motor\_2\_Direction
  - azatrax::SL2, [917](#)
- Motor\_2\_State
  - azatrax::SL2, [917](#)
- motor\_2\_direction
  - azatrax::SL2::status1\_union, [962](#)
- motor\_2\_state
  - azatrax::SL2::status1\_union, [962](#)
- move
  - CTCPanel::CTCPanel, [384](#)
- MovementsThisSession
  - FCFSupport::Car, [251](#)
- moves
  - FCFSupport::Car, [261](#)
- mtidetail
  - lcc::CANGridConnect, [214](#)
  - lcc::OpenLCBOverTcp, [706](#)
- mtiheader
  - lcc::CANGridConnect, [214](#)
  - lcc::MTIDetail, [639](#)
- MyDivision
  - FCFSupport::Station, [944](#)
- MyMirror
  - FCFSupport::Industry, [531](#)
- MyProduct
  - azatrax::Azatrax, [176](#)
- MyProductId
  - azatrax::Azatrax, [176](#)
- myProductId
  - azatrax::Azatrax, [181](#)
- mySerialNumber
  - azatrax::Azatrax, [181](#)
- MyStation
  - FCFSupport::Industry, [531](#)
- myalias
  - lcc::CanAlias, [203](#)
- mycanalias
  - lcc::CANGridConnect, [214](#)
- NCEModule, [65](#)
- NCE
  - nce::NCE, [648](#)
- NIDPATTERN
  - lcc::CANGridConnect, [215](#)
  - lcc::OpenLCBOverTcp, [706](#)
- NOP
  - nce::NCE, [655](#)
- NScale
  - MRRXtrkCad.tab.h, [1310](#)
- Name
  - FCFSupport::Division, [412](#)
  - FCFSupport::Industry, [531](#)
  - FCFSupport::Owner, [732](#)
  - FCFSupport::Station, [944](#)
  - FCFSupport::Train, [1212](#)
  - TTSupport::Cab, [194](#)
  - TTSupport::Station, [950](#)
  - TTSupport::StorageTrack, [988](#)
  - TTSupport::TimeTableSystem, [1147](#)

TTSupport::Train, 1199  
 name  
   FCFSupport::Division, 414  
   FCFSupport::Industry, 536  
   FCFSupport::Owner, 733  
   FCFSupport::Station, 946  
   FCFSupport::Train, 1218  
   lcc::ConfigOptions, 314  
   Parsers::TrackGraph::NodeValues, 676  
   TTSupport::Cab, 196  
   TTSupport::Station, 952  
   TTSupport::StorageTrack, 992  
   TTSupport::TimeTableSystem, 1152  
   TTSupport::Train, 1204  
 NameOfNode  
   Parsers::LayoutFile, 559  
   Parsers::TrackGraph, 1188  
 NamedIndirectObjectMap  
   FCFSupport::PDFFileStructures, 91  
 nce, 128  
   AccessoryNumber, 129  
   AspectBits, 130  
   CSAddress, 130  
   CabNumber, 130  
   ConsistAddress, 130  
   Direction, 133  
   EchoMode, 130  
   ErrorMessage, 134  
   Forward, 134  
   Hours, 131  
   LCDMessage16, 131  
   LCDMessage8, 131  
   LocoAddress, 131  
   MacroNumber, 131  
   Minutes, 132  
   MomentumLevel, 132  
   RAMData, 132  
   RAMData8, 132  
   RawPacket, 132  
   RawTrackPacket, 132  
   Reverse, 134  
   S128, 134  
   S14, 134  
   S28, 134  
   ScaleClockRatio, 133  
   Speed128, 133  
   Speed28, 133  
   SpeedMode, 134  
   UByte, 133  
 nce::NCE, 645  
   \_explodechars, 649  
   \_readbyte, 649  
   \_readevent, 650  
   \_readresponse, 650  
   \_sendMessageAndReturnResponse, 650  
   \_timeout, 672  
   \_timeoutevent, 650  
   \_transmit, 650  
   ~NCE, 649  
   AccessoryDecoderOperation, 651  
   AddLeadLocomotiveToMultiUnit, 651  
   AddLocomotiveToMultiUnit, 652  
   AddRearLocomotiveToMultiUnit, 652  
   AssignLoco, 652  
   ChangeMomentumLevel, 653  
   DequeuePacket, 653  
   DisableMain, 654  
   Dummy, 654  
   EnableMain, 654  
   ExecuteMacro, 654  
   HardReset, 655  
   MacroCommand, 655  
   NCE, 648  
   NOP, 655  
   NormalMode, 655  
   NumberOfBytesReturned, 672  
   OperatingModeAccessoryProgrammingByteMode↔  
     Write, 656  
   OperatingModeProgrammingByteModeWrite, 656  
   ProgramMode, 657  
   ReadCVInDirectMode, 657  
   ReadCVInPagedMode, 657  
   ReadFromRAM, 658  
   ReadOneByteFromRAM, 658  
   ReadRegister, 658  
   RemoveLocomotiveFromMultiUnit, 659  
   ReturnAuxiliaryInputUnit, 659  
   ReturnAuxiliaryInputUnitShortForm, 660  
   ReturnClock, 660  
   SetBinaryCommandEchoMode, 660  
   SetCabBusAddressOfUSBBoard, 661  
   SetClock, 661  
   SetClockFormat, 661  
   SetClockRatio, 662  
   SetLocoSpeedMode, 664  
   SetLocomotiveFunctionsGroup1, 662  
   SetLocomotiveFunctionsGroup2, 663  
   SetLocomotiveFunctionsGroup3, 663  
   SetLocomotiveSpeedAndDirection, 664  
   SetSignalAspect, 665  
   SoftReset, 665  
   SoftwareVersion, 665  
   StartClock, 665  
   StopClock, 666  
   ttyfd, 672  
   Write4BytesToRAM, 666  
   Write8BytesToRAM, 666  
   WriteCVInDirectMode, 667

- WriteCVInPagedMode, [667](#)
- WriteLCDLine3, [668](#)
- WriteLCDLine4, [668](#)
- WriteLCDRightLine2, [669](#)
- WriteOneByteToRAM, [669](#)
- WriteRAWPacket, [669](#)
- WriteRAWTrackPacket, [670](#)
- WriteRegister, [670](#)
- WriteToRAM, [671](#)
- WriteTwoBytesToRAM, [671](#)
- needPage
  - FCFSupport::PDFPrinterDevice, [785](#)
- needPageHeader
  - FCFSupport::PostScriptPrinterDevice, [798](#)
- NetworkOffline
  - CTIAcela, [52](#)
- NetworkOnline
  - CTIAcela, [52](#)
- networkonline
  - CTIAcela, [62](#)
- NewCreateTimeTable
  - TimeTableSystemTcl, [30](#)
- NewPage
  - FCFSupport::LQ24PrinterDevice, [600](#)
  - FCFSupport::PDFPrinterDevice, [781](#)
  - FCFSupport::PostScriptPrinterDevice, [794](#)
  - FCFSupport::PrinterDevice, [805](#)
  - FCFSupport::TextPrinterDevice, [1113](#)
- Next
  - Parsers::IntegerList, [544](#)
- next
  - Parsers::BezierBody, [185](#)
  - Parsers::CornuBody, [342](#)
  - Parsers::IntegerList, [545](#)
  - Parsers::TrackBody, [1163](#)
  - Parsers::TurnoutBody, [1230](#)
- NextShift
  - FCFSupport::System, [1053](#)
- NextSwitchListForCarAndIndustry
  - FCFSupport::SwitchList, [1008](#)
- nextlink
  - HTMLHelp::HTMLHelp, [507](#)
- Nibble
  - xpressnet::AccessoryDecoderInformation, [164](#)
- nibble
  - xpressnet, [155](#)
- NibbleCode
  - xpressnet, [158](#)
- nid
  - lcc, [123](#)
- nidLEntry
  - lcc::CANGridConnectOverCANSocket, [219](#)
  - lcc::CANGridConnectOverTcp, [224](#)
  - lcc::CANGridConnectOverUSBSerial, [228](#)
  - lcc::OpenLCBOverTcp, [706](#)
- nidMap
  - lcc::CANGridConnect, [214](#)
- nidlist
  - lcc::CanAlias, [203](#)
  - lcc::OpenLCBNode, [694](#)
- Node
  - Parsers::TrackGraph, [1175](#)
- nodeStack
  - ParseXML, [757](#)
- NodeTurnoutGraphic
  - Parsers::LayoutFile, [560](#)
  - Parsers::TrackGraph, [1188](#)
- NodeTurnoutRoutelist
  - Parsers::LayoutFile, [560](#)
  - Parsers::TrackGraph, [1189](#)
- NodeType
  - Parsers::TrackGraph, [1175](#)
- NodeValues
  - Parsers::TrackGraph::NodeValues, [674](#)
- nodeid
  - lcc::ConfigOptions, [314](#)
- nodes
  - Parsers::TrackGraph, [1195](#)
- none
  - Parsers::TrackGraph, [1195](#)
- NormalActionScript
  - Parsers::LayoutFile, [560](#)
  - Parsers::TrackGraph, [1189](#)
- NormalMode
  - nce::NCE, [655](#)
- normalactionscript
  - Parsers::TrackGraph::NodeValues, [676](#)
- Note
  - TTSupport::Stop, [980](#)
  - TTSupport::TimeTableSystem, [1147](#)
  - TTSupport::Train, [1199](#)
- notes
  - TTSupport::Stop, [984](#)
  - TTSupport::TimeTableSystem, [1152](#)
  - TTSupport::Train, [1205](#)
- NumEdges
  - Parsers::LayoutFile, [561](#)
  - Parsers::TrackGraph, [1189](#)
- numRoutelists
  - Parsers::TurnoutRoutelist, [1242](#)
- numSegments
  - Parsers::TurnoutGraphic, [1241](#)
- Number
  - FCFSupport::Car, [251](#)
  - TTSupport::Train, [1200](#)
- number
  - FCFSupport::Car, [262](#)
  - TTSupport::Train, [1205](#)

- numberCars
  - FCFSupport::System, [1103](#)
- NumberOfBytesReturned
  - nce::NCE, [672](#)
- NumberOfCabs
  - TTSupport::TimeTableSystem, [1147](#)
- NumberOfCars
  - FCFSupport::Industry, [531](#)
  - FCFSupport::System, [1054](#)
- NumberOfDivisions
  - FCFSupport::System, [1054](#)
- NumberOfFeedbackElements
  - xpressnet::AccessoryDecoderInformation, [164](#)
- NumberOfHeads
  - Parsers::LayoutFile, [560](#)
  - Parsers::TrackGraph, [1189](#)
- NumberOfIndustries
  - FCFSupport::Station, [944](#)
  - FCFSupport::System, [1054](#)
- NumberOfNotes
  - TTSupport::Stop, [980](#)
  - TTSupport::TimeTableSystem, [1147](#)
  - TTSupport::Train, [1200](#)
- NumberOfOpenDevices
  - azatrax::Azatrax, [177](#)
- NumberOfOrders
  - FCFSupport::Train, [1213](#)
- NumberOfStations
  - FCFSupport::Division, [412](#)
  - FCFSupport::System, [1054](#)
  - TTSupport::TimeTableSystem, [1147](#)
- NumberOfStops
  - FCFSupport::Train, [1213](#)
  - TTSupport::Train, [1200](#)
- NumberOfStorageTracks
  - TTSupport::Station, [950](#)
- NumberOfTrains
  - FCFSupport::System, [1054](#)
  - TTSupport::TimeTableSystem, [1148](#)
- NumberStyle
  - FCFSupport::PDFFileStructures::PageLabel↔  
Dictionary, [739](#)
- numheads
  - Parsers::TrackGraph::NodeValues, [676](#)
- nums
  - FCFSupport::PDFFileStructures::PageLabelTree,  
[745](#)
- numtoolbars
  - mainwindow, [619](#)
- OPENLCBFRAME\_MASK
  - lcc::CANHeader, [231](#)
- OPENLCBFRAME\_SHIFT
  - lcc::CANHeader, [231](#)
- OScale
  - MRRXtrkCad.tab.h, [1311](#)
- objectMap
  - FCFSupport::PDFFileStructures::CrossReference↔  
Table, [359](#)
- ObjectNumber
  - FCFSupport::PDFFileStructures::IndirectObject, [518](#)
- objectNumber
  - FCFSupport::PDFFileStructures::IndirectObject, [520](#)
- objectTable
  - FCFSupport::PDFFileStructures::CrossReference↔  
Table, [361](#)
- objectlist
  - CTCPanel::CTCPanel, [385](#)
- Objects
  - CTCPanel::CTCPanel, [389](#)
- occupations
  - TTSupport::StorageTrack, [992](#)
- Occupied
  - TTSupport::Occupied, [680](#)
- OccupiedMap
  - Station, [34](#)
- OffScript
  - Parsers::LayoutFile, [561](#)
  - Parsers::TrackGraph, [1189](#)
- offscript
  - Parsers::TrackGraph::NodeValues, [676](#)
- OkToMirrorP
  - FCFSupport::Car, [251](#)
- OldCreateTimeTable
  - TimeTableSystemTcl, [30](#)
- olddatagramhandler
  - lcc::ConfigMemory, [310](#)
  - lcc::ConfigurationEditor, [338](#)
- oldgeneralmessagehandler
  - lcc::ConfigMemory, [310](#)
- OnDuty
  - FCFSupport::Train, [1213](#)
- OnScript
  - Parsers::LayoutFile, [561](#)
  - Parsers::TrackGraph, [1190](#)
- onduty
  - FCFSupport::Train, [1218](#)
- oneColumnWidthFraction
  - FCFSupport::LQ24PrinterDevice, [604](#)
  - PDFPrinter.h, [1293](#)
- OnlineP
  - CTIAcela, [52](#)
- onscript
  - Parsers::TrackGraph::NodeValues, [677](#)
- Opcodes
  - CTIAcela, [62](#)
- open
  - lcc::EventLog, [440](#)



OpenDevice  
     azatrax::Azatrax, [177](#)  
 OpenLCBMessage  
     lcc::OpenLCBMessage, [685](#)  
 OpenLCBNode  
     lcc::OpenLCBNode, [689](#)  
 OpenLCBOverTcp  
     lcc::OpenLCBOverTcp, [702](#)  
 OpenPrinter  
     FCFSupport::LQ24PrinterDevice, [600](#)  
     FCFSupport::PDFPrinterDevice, [781](#)  
     FCFSupport::PostScriptPrinterDevice, [795](#)  
     FCFSupport::PrinterDevice, [805](#)  
     FCFSupport::TextPrinterDevice, [1113](#)  
 openport  
     CmriSupport::CmriNode, [288](#)  
 OperatingMode  
     azatrax::MRD, [626](#)  
 operatingMode  
     azatrax::Azatrax::StateDataPacket, [939](#)  
 OperatingMode\_Type  
     azatrax::MRD, [623](#)  
 OperatingModeAccessoryProgrammingByteModeWrite  
     nce::NCE, [656](#)  
 OperatingModeProgrammingBitModeWrite  
     xpressnet::XPressNet, [1265](#)  
 OperatingModeProgrammingByteModeWrite  
     nce::NCE, [656](#)  
     xpressnet::XPressNet, [1266](#)  
 operator!=  
     Parsers::TrackGraph::Transform2D, [1226](#)  
 operator<  
     FCFSupport::PathName, [763](#)  
     TTSupport::PathName, [771](#)  
     TTSupport::TimeRange, [1126](#)  
 operator<<  
     FCFSupport, [89](#)  
     FCFSupport::SwitchList, [1011](#)  
     Parsers::IntegerList, [545](#)  
     Parsers::TrackBody, [1162](#)  
     Parsers::TrackBodyElt, [1166](#)  
     Parsers::TrackGraph, [1193](#)  
 operator<=  
     FCFSupport::PathName, [763](#)  
     TTSupport::PathName, [771](#)  
     TTSupport::TimeRange, [1127](#)  
 operator>  
     FCFSupport::PathName, [764](#)  
     TTSupport::PathName, [773](#)  
     TTSupport::TimeRange, [1128](#)  
 operator>=  
     FCFSupport::PathName, [765](#)  
     TTSupport::PathName, [773](#)  
     TTSupport::TimeRange, [1128](#)  
 operator\*  
     Parsers::TrackGraph::Transform2D, [1226](#)  
 operator()  
     TTSupport::eqstr, [434](#)  
     TTSupport::hash, [479](#)  
 operator+  
     FCFSupport::PathName, [761](#), [762](#)  
     TTSupport::PathName, [770](#)  
 operator+=  
     FCFSupport::PathName, [762](#)  
     TTSupport::PathName, [771](#)  
 operator=  
     FCFSupport::Car, [252](#)  
     FCFSupport::CarGroup, [266](#)  
     FCFSupport::CarType, [270](#)  
     FCFSupport::Division, [412](#)  
     FCFSupport::Industry, [532](#)  
     FCFSupport::Owner, [733](#)  
     FCFSupport::PathName, [763](#), [764](#)  
     FCFSupport::Station, [944](#)  
     FCFSupport::SwitchListElement, [1016](#)  
     FCFSupport::Train, [1213](#)  
     TTSupport::Cab, [194](#)  
     TTSupport::Occupied, [681](#)  
     TTSupport::PathName, [772](#)  
     TTSupport::Station, [951](#)  
     TTSupport::StationTimes, [957](#)  
     TTSupport::Stop, [980](#)  
     TTSupport::StorageTrack, [988](#)  
     TTSupport::TimeRange, [1127](#)  
 operator==  
     FCFSupport::PathName, [764](#)  
     Parsers::TrackGraph::Transform2D, [1226](#)  
     TTSupport::PathName, [772](#)  
     TTSupport::TimeRange, [1127](#)  
 operator[]  
     FCFSupport::SwitchList, [1009](#)  
 OptionHashMap  
     TTSupport, [150](#)  
 Order  
     FCFSupport::Train, [1214](#)  
 orders  
     FCFSupport::Train, [1219](#)  
 OrdersFile  
     FCFSupport::System, [1055](#)  
 ordersFile  
     FCFSupport::System, [1103](#)  
 originYard  
     FCFSupport::System, [1104](#)  
 OrigX  
     Parsers::LayoutFile, [562](#)  
     Parsers::TrackGraph, [1190](#)  
 origx  
     Parsers::TrackGraph::NodeValues, [677](#)



- OrigY
  - Parsers::LayoutFile, [562](#)
  - Parsers::TrackGraph, [1190](#)
- origy
  - Parsers::TrackGraph::NodeValues, [677](#)
- Other
  - xpressnet, [158](#)
- OtherCarOkForTrain
  - FCFSupport::System, [1055](#)
- out
  - linuxgpio, [127](#)
- OutputRelayInputControl
  - azatrax::SL2, [918](#)
  - azatrax::SR4, [932](#)
- outputbuffer
  - CmriSupport::CmriNode, [290](#)
- Outputs
  - cmri::CMri, [281](#)
- outputs
  - CmriSupport::CmriNode, [289](#)
- OvalButton
  - OvalWidgets::OvalButton, [710](#)
- OvalLabel
  - OvalWidgets, [139](#)
- OvalRoundCornerRectangle
  - OvalWidgets::OvalRoundCornerRectangle, [713](#)
- OvalScale
  - OvalWidgets::OvalScale, [715](#)
- OvalSlider
  - OvalWidgets::OvalSlider, [720](#)
- OvalScrollBar
  - OvalWidgets::OvalScrollBar, [725](#)
- OvalWidgets, [135](#)
  - \_ConfigureFont, [136](#)
  - \_ConfigureText, [136](#)
  - \_UnderSplit, [137](#)
  - \_VerifyFont, [137](#)
  - \_VerifyIntegerOrEmpty, [137](#)
  - ~OvalLabel, [140](#)
  - canvas, [140](#)
  - ColorFillOption, [138](#)
  - ColorOptionMethods, [138](#)
  - ColorOutlineOption, [138](#)
  - CommonValidateMethods, [139](#)
  - FontFamily, [139](#)
  - HBar, [140](#)
  - OvalLabel, [139](#)
  - SquareEndOptions, [139](#)
  - VBar, [141](#)
  - XYWH, [140](#)
- OvalWidgets::OvalButton, [709](#)
  - \_ConfigureText, [711](#)
  - ~OvalButton, [711](#)
  - canvas, [712](#)
  - invoke, [711](#)
  - OvalButton, [710](#)
- OvalWidgets::OvalRoundCornerRectangle, [712](#)
  - ~OvalRoundCornerRectangle, [713](#)
  - canvas, [713](#)
  - OvalRoundCornerRectangle, [713](#)
- OvalWidgets::OvalScale, [713](#)
  - \_BaseRect, [716](#)
  - \_ConfigureText, [716](#)
  - \_ConfigureWL, [716](#)
  - \_ConfigureXY, [717](#)
  - \_MoveThumb, [717](#)
  - \_value, [718](#)
  - ~OvalScale, [716](#)
  - canvas, [718](#)
  - get, [717](#)
  - OvalScale, [715](#)
  - set, [718](#)
- OvalWidgets::OvalSlider, [718](#)
  - \_ConfigureL, [721](#)
  - \_ConfigureText, [721](#)
  - \_MoveThumb, [721](#)
  - \_VerifyBitmap, [722](#)
  - \_value, [723](#)
  - canvas, [723](#)
  - get, [722](#)
  - OvalSlider, [720](#)
  - set, [722](#)
- OvalWidgets::OvalScrollBar, [723](#)
  - \_BaseRect, [725](#)
  - \_Command, [726](#)
  - \_ConfigureWL, [726](#)
  - \_ConfigureXY, [726](#)
  - \_MoveThumb, [727](#)
  - \_lastSet, [729](#)
  - ~OvalScrollBar, [725](#)
  - canvas, [729](#)
  - delta, [727](#)
  - fraction, [727](#)
  - get, [728](#)
  - identify, [728](#)
  - OvalScrollBar, [725](#)
  - resize, [728](#)
  - set, [729](#)
- Owner
  - FCFSupport::Owner, [731](#)
- owner
  - FCFSupport::Car, [262](#)
- OwnerMap
  - FCFSupport, [87](#)
- owners
  - FCFSupport::System, [1104](#)
- OwnersFile
  - FCFSupport::System, [1055](#)

- ownersFile
  - FCFSupport::System, [1104](#)
- PDFNameArray
  - FCFSupport::PDFFileStructures::PDFNameArray, [777](#)
- PDFPrinter.h
  - oneColumnWidthFraction, [1293](#)
- PDFPrinterDevice
  - FCFSupport::PDFPrinterDevice, [779](#)
- PDFPrinterSupport.h
  - asctime\_r, [1295](#)
  - localtime\_r, [1295](#)
- PDFStream
  - FCFSupport::PDFFileStructures::PDFStream, [787](#)
- PDFStreamVector
  - FCFSupport::PDFFileStructures, [92](#)
- PI2
  - GRSupport, [102](#)
- PIEngineering
  - RaildriverIO, [837](#)
- PRIORITY\_MASK
  - lcc::MTIDetail, [640](#)
- PRIORITY\_SHIFT
  - lcc::MTIDetail, [640](#)
- PSQuote
  - FCFSupport::PostScriptPrinterDevice, [795](#)
- pack4
  - CTIAcela, [53](#)
- pack8
  - CTIAcela, [53](#)
- PacketCount
  - azatrax::Azatrax, [177](#)
- packetCount
  - azatrax::Azatrax::StateDataPacket, [939](#)
- Page
  - FCFSupport::PDFFileStructures::Page, [735](#)
- PageLabelDictionary
  - FCFSupport::PDFFileStructures::PageLabelDictionary, [739](#)
- PageLabelDictionaryNumMap
  - FCFSupport::PDFFileStructures, [92](#)
- PageLabelTree
  - FCFSupport::PDFFileStructures::PageLabelTree, [742](#)
- PageLabelTreeKidVector
  - FCFSupport::PDFFileStructures, [92](#)
- PageSize
  - FCFSupport::PrinterDevice, [802](#)
- pageSize
  - FCFSupport::PrinterDevice, [809](#)
- PageTree
  - FCFSupport::PDFFileStructures::Page, [736](#)
  - FCFSupport::PDFFileStructures::PageTree, [746](#)
- pageTreeRoot
  - FCFSupport::PDFPrinterDevice, [785](#)
- PagedModeCVRead
  - xpressnet::XPressNet, [1266](#)
- PagedModeCVWrite
  - xpressnet::XPressNet, [1266](#)
- pagenodes
  - FCFSupport::PDFFileStructures::PageTree, [749](#)
- pages
  - FCFSupport::PDFFileStructures::CatalogDictionary, [275](#)
  - FCFSupport::PostScriptPrinterDevice, [798](#)
  - ScrollTabNotebook, [877](#)
- pages\_opts
  - ScrollTabNotebook, [877](#)
- PanedWindow, [141](#)
- panes
  - HTMLHelp::HTMLHelp, [512](#)
- panewindow
  - mainwindow, [619](#)
- parent
  - FCFSupport::PDFFileStructures::Page, [737](#)
  - FCFSupport::PDFFileStructures::PageTree, [749](#)
  - lcc::CANGridConnect, [215](#)
- Parse
  - Parsers::LayoutFile, [562](#)
  - Parsers::ParseFile, [751](#)
- ParseError
  - Parsers::LayoutFile, [562](#)
  - Parsers::ParseFile, [751](#)
- ParseFile
  - Parsers::ParseFile, [751](#)
- ParseXML, [753](#)
  - \_characterdata, [756](#)
  - \_elementend, [756](#)
  - \_elementstart, [756](#)
  - displayTree, [757](#)
  - nodeStack, [757](#)
  - ParseXML, [754](#)
  - rootnode, [758](#)
  - validate, [757](#)
- parser
  - Parsers::LayoutFile, [564](#)
- ParserClasses, [25](#)
  - tcl\_socketpair, [25](#)
- Parsers, [141](#)
- Parsers::BezierBody, [182](#)
  - BezierBody, [183](#)
  - BezierBodyElt, [185](#)
  - BezierEnds, [183](#)
  - BezierSegmentCount, [183](#)
  - CleanUpBezierBody, [184](#)
  - CleanUpElement, [184](#)
  - ConsBezierBody, [184](#)

- Element, [184](#)
- element, [185](#)
- next, [185](#)
- TrackGraph, [185](#)
- Parsers::BezierBodyElt, [185](#)
  - ~BezierBodyElt, [188](#)
  - ang0, [190](#)
  - ang1, [190](#)
  - BezierBody, [190](#)
  - BezierBodyElt, [188](#)
  - BezierBodyEltType, [187](#)
  - GetCurveSegment, [188](#)
  - GetStraightSegment, [188](#)
  - InitTSegId, [188](#)
  - MakeCurveSegment, [189](#)
  - MakeStraightSegment, [189](#)
  - MakeTrackEnd, [189](#)
  - pos1, [190](#)
  - pos2, [191](#)
  - radius, [191](#)
  - segCount, [191](#)
  - segmentId, [191](#)
  - theEnd, [191](#)
  - TheType, [189](#)
  - theType, [192](#)
  - TrackGraph, [190](#)
- Parsers::BezierBodyElt::Pos, [790](#)
  - x, [790](#)
  - y, [790](#)
- Parsers::CornuBody, [339](#)
  - CleanUpCornuBody, [340](#)
  - CleanUpElement, [340](#)
  - ConcatCornuBody, [340](#)
  - ConsCornuBody, [341](#)
  - CornuBody, [340](#)
  - CornuBodyElt, [342](#)
  - CornuEnds, [341](#)
  - CornuSegmentCount, [341](#)
  - Element, [341](#)
  - element, [342](#)
  - next, [342](#)
  - TrackGraph, [342](#)
- Parsers::CornuBodyElt, [342](#)
  - ~CornuBodyElt, [345](#)
  - ang0, [347](#)
  - ang1, [347](#)
  - CornuBody, [347](#)
  - CornuBodyElt, [345](#)
  - CornuBodyEltType, [344](#)
  - GetCurveSegment, [345](#)
  - GetStraightSegment, [345](#)
  - InitTSegId, [345](#)
  - MakeCurveSegment, [346](#)
  - MakeStraightSegment, [346](#)
  - MakeTrackEnd, [346](#)
  - pos1, [347](#)
  - pos2, [348](#)
  - radius, [348](#)
  - segCount, [348](#)
  - segmentId, [348](#)
  - theEnd, [348](#)
  - TheType, [346](#)
  - theType, [349](#)
  - TrackGraph, [347](#)
- Parsers::CornuBodyElt::Pos, [791](#)
  - x, [791](#)
  - y, [791](#)
- Parsers::IntegerList, [542](#)
  - CleanUpIntegerList, [543](#)
  - CopyList, [543](#)
  - Element, [543](#)
  - ElementP, [544](#)
  - iElt, [545](#)
  - IntAppend, [544](#)
  - IntegerList, [543](#)
  - Next, [544](#)
  - next, [545](#)
  - operator<<, [545](#)
  - TurnoutBodyElt, [545](#)
- Parsers::LayoutFile, [550](#)
  - ~LayoutFile, [553](#)
  - Angle, [553](#)
  - CompressGraph, [556](#)
  - CompressedEdgeCount, [554](#)
  - CompressedEdgeLength, [554](#)
  - CompressedEdgeNode, [554](#)
  - CompressedGraphCircleLayout, [554](#)
  - CompressedGraphKamadaKawaiSpring, [555](#)
  - CompressedGraphKruskalMinimumSpanningTree, [555](#)
  - CompressedGraphPrimMinimumSpanningTree, [555](#)
  - CompressedNodePositionX, [555](#)
  - CompressedNodePositionY, [556](#)
  - CompressedNodeSegments, [556](#)
  - EdgeIndex, [557](#)
  - EdgeLength, [557](#)
  - EdgeA, [556](#)
  - EdgeX, [557](#)
  - EdgeY, [557](#)
  - Emit, [558](#)
  - Heads, [558](#)
  - HighestNode, [558](#)
  - IsCompressed, [558](#)
  - IsCompressedNode, [559](#)
  - IsNodeP, [559](#)
  - LayoutFile, [553](#)
  - LengthOfNode, [559](#)
  - LowestNode, [559](#)

- NameOfNode, 559
- NodeTurnoutGraphic, 560
- NodeTurnoutRoutelist, 560
- NormalActionScript, 560
- NumEdges, 561
- NumberOfHeads, 560
- OffScript, 561
- OnScript, 561
- OrigX, 562
- OrigY, 562
- Parse, 562
- ParseError, 562
- parser, 564
- ReverseActionScript, 563
- Roots, 563
- SenseScript, 563
- SignalAspects, 563
- trackGraph, 565
- TrackList, 564
- TurnoutNumber, 564
- TypeOfNode, 564
- Parsers::MRRXtrkCad, 629
  - ~MRRXtrkCad, 632
  - CurrentScale, 633
  - fieldflag, 633
  - lookup\_word, 632
  - MRRXtrkCad, 632
  - scanEol, 633
  - scanToEND, 634
  - YY\_MRRXtrkCad\_ENUM\_TOKEN, 630
  - yychar, 634
  - yydebug, 634
  - yyerror, 632
  - yyerror1, 632
  - yylex, 633
  - yyloc, 634
  - yylval, 634
  - yynerrs, 634
  - yyparse, 633
- Parsers::ParseFile, 749
  - ~ParseFile, 751
  - bufferize, 752
  - errorstream, 752
  - fp, 752
  - line\_buffer, 752
  - lp, 753
  - Parse, 751
  - ParseError, 751
  - ParseFile, 751
  - ProcessFile, 751
  - source\_file, 753
  - source\_line, 753
  - SourceFile, 752
- Parsers::RouteVec, 851
  - posList, 851
  - positionName, 851
  - routeLength, 851
- Parsers::SegPos, 878
  - x, 879
  - y, 879
- Parsers::SegVector, 879
  - ang0, 881
  - ang1, 881
  - angle, 881
  - ePos1, 881
  - ePos2, 881
  - gPos1, 881
  - gPos2, 882
  - GrType, 880
  - L, 882
  - len0, 882
  - len1, 882
  - length, 882
  - R, 882
  - radius, 883
  - tgType, 883
- Parsers::TrackBody, 1160
  - ~TrackBody, 1161
  - AppendTrackBodyElt, 1162
  - ConsTrackBody, 1162
  - element, 1163
  - next, 1163
  - operator<<, 1162
  - TrackBody, 1161
  - TrackBodyLength, 1162
  - TrackGraph, 1163
- Parsers::TrackBodyElt, 1163
  - ~TrackBodyElt, 1165
  - a, 1166
  - ConnectedTrackEnd, 1165
  - index, 1166
  - operator<<, 1166
  - TrackBodyElt, 1164
  - TrackGraph, 1166
  - UnConnectedTrackEnd, 1165
  - x, 1166
  - y, 1167
- Parsers::TrackGraph, 1167
  - ~TrackGraph, 1176
  - AddNewNode, 1176
  - Angle, 1176
  - backpointers, 1193
  - c\_idMap, 1194
  - c\_nodes, 1194
  - c\_roots, 1194
  - circleLayoutP, 1194
  - CompressGraph, 1179
  - compressed\_edge\_exists, 1177

- CompressedEdgeCount, [1177](#)
- CompressedEdgeLength, [1177](#)
- CompressedEdgeNode, [1177](#)
- CompressedEdgePair, [1173](#)
- CompressedEdgePairVector, [1173](#)
- CompressedGraph, [1173](#)
- CompressedGraphCircleLayout, [1177](#)
- CompressedGraphKamadaKawaiSpring, [1178](#)
- CompressedGraphKruskalMinimumSpanningTree, [1178](#)
- CompressedGraphPrimMinimumSpanningTree, [1178](#)
- CompressedIdNodeMap, [1174](#)
- CompressedNode, [1174](#)
- CompressedNodePositionX, [1178](#)
- CompressedNodePositionY, [1179](#)
- CompressedNodeSegments, [1179](#)
- CompressedNodeVector, [1174](#)
- compressedP, [1194](#)
- computeHeads, [1179](#)
- ComputeRouteLength, [1179](#)
- DeleteTurnoutGraphic, [1180](#)
- DeleteTurnoutRouteList, [1180](#)
- EdgeIndex, [1180](#)
- EdgeLength, [1180](#)
- EdgeA, [1180](#)
- EdgeX, [1181](#)
- EdgeY, [1181](#)
- FindBlock, [1181](#)
- FindNode, [1182](#)
- Graph, [1174](#)
- Heads, [1182](#)
- heads, [1194](#)
- HighestNode, [1182](#)
- idMap, [1195](#)
- IdNodeMap, [1174](#)
- InsertBezierTrack, [1182](#)
- InsertBlock, [1182](#)
- insertCompressedNode, [1183](#)
- InsertControl, [1183](#)
- InsertCornuTrack, [1183](#)
- InsertCurveTrack, [1183](#)
- InsertJointTrack, [1184](#)
- InsertSensor, [1184](#)
- InsertSignal, [1184](#)
- InsertStraightTrack, [1184](#)
- InsertSwitchMotor, [1185](#)
- InsertTurnOut, [1185](#)
- InsertTurnTable, [1185](#)
- IsCompressed, [1185](#)
- IsCompressedNode, [1186](#)
- IsNodeP, [1186](#)
- IsNone, [1186](#)
- KamadaKawaiSpringLayoutP, [1195](#)
- LengthOfCurve, [1186](#)
- LengthOfJoint, [1187](#)
- LengthOfNode, [1187](#)
- LengthOfStraight, [1187](#)
- LowestNode, [1187](#)
- MakeTurnoutGraphic, [1188](#)
- MakeTurnoutRouteList, [1188](#)
- NameOfNode, [1188](#)
- Node, [1175](#)
- NodeTurnoutGraphic, [1188](#)
- NodeTurnoutRoutelist, [1189](#)
- NodeType, [1175](#)
- nodes, [1195](#)
- none, [1195](#)
- NormalActionScript, [1189](#)
- NumEdges, [1189](#)
- NumberOfHeads, [1189](#)
- OffScript, [1189](#)
- OnScript, [1190](#)
- operator<<, [1193](#)
- OrigX, [1190](#)
- OrigY, [1190](#)
- ReverseActionScript, [1191](#)
- Roots, [1191](#)
- RotationUnit, [1175](#)
- SenseScript, [1191](#)
- SignalAspects, [1191](#)
- tr\_rotate, [1191](#)
- tr\_scale, [1192](#)
- tr\_translate, [1192](#)
- TrackGraph, [1176](#)
- TrackList, [1192](#)
- traversePrimMST, [1192](#)
- TurnoutNumber, [1193](#)
- TypeOfNode, [1193](#)
- valid\_heads, [1195](#)
- Parsers::TrackGraph::CompressedEdgeValues, [299](#)
  - CompressedEdgeValues, [300](#)
  - length, [300](#)
- Parsers::TrackGraph::CompressedNodeValues, [300](#)
  - CompressedNodeValues, [301](#)
  - FindSegmentIndex, [301](#)
  - id, [302](#)
  - position, [302](#)
  - rawnode, [302](#)
  - segments, [302](#)
- Parsers::TrackGraph::EdgeValues, [428](#)
  - a, [429](#)
  - EdgeValues, [429](#)
  - index, [429](#)
  - length, [429](#)
  - x, [429](#)
  - y, [430](#)
- Parsers::TrackGraph::NodeValues, [673](#)
  - angle, [675](#)

- aspectlist, [675](#)
- Cleanup, [675](#)
- id, [676](#)
- length, [676](#)
- name, [676](#)
- NodeValues, [674](#)
- normalactionscrip, [676](#)
- numheads, [676](#)
- offscript, [676](#)
- onscript, [677](#)
- origx, [677](#)
- origy, [677](#)
- reverseactionscrip, [677](#)
- sensescrip, [677](#)
- tgr, [677](#)
- tpo, [678](#)
- tracklist, [678](#)
- turnoutnumber, [678](#)
- type, [678](#)
- Parsers::TrackGraph::Point, [788](#)
  - x, [788](#)
  - y, [788](#)
- Parsers::TrackGraph::Transform2D, [1223](#)
  - Apply, [1225](#)
  - Determinant, [1225](#)
  - FUZZ, [1226](#)
  - Inverse, [1225](#)
  - matrix, [1227](#)
  - Minor, [1225](#)
  - operator!=, [1226](#)
  - operator\*, [1226](#)
  - operator==, [1226](#)
  - Transform2D, [1224](#)
- Parsers::TurnoutBody, [1227](#)
  - CleanUpElement, [1228](#)
  - CleanUpTurnoutBody, [1228](#)
  - ConsTurnoutBody, [1229](#)
  - Element, [1229](#)
  - element, [1230](#)
  - next, [1230](#)
  - TrackGraph, [1230](#)
  - TurnoutBody, [1228](#)
  - TurnoutBodyElt, [1230](#)
  - TurnoutEnds, [1229](#)
  - TurnoutRouteCount, [1229](#)
  - TurnoutSegmentCount, [1229](#)
- Parsers::TurnoutBodyElt, [1231](#)
  - ~TurnoutBodyElt, [1233](#)
  - ang0, [1237](#)
  - ang1, [1237](#)
  - GetTurnoutCurveSegment, [1233](#)
  - GetTurnoutJointSegment, [1234](#)
  - GetTurnoutRoute, [1234](#)
  - GetTurnoutStraightSegment, [1234](#)
  - InitTSegId, [1234](#)
  - L, [1237](#)
  - MakeTurnoutCurveSegment, [1235](#)
  - MakeTurnoutEnd, [1235](#)
  - MakeTurnoutJointSegment, [1235](#)
  - MakeTurnoutRoute, [1235](#)
  - MakeTurnoutStraightSegment, [1236](#)
  - pos1, [1237](#)
  - pos2, [1237](#)
  - R, [1238](#)
  - radius, [1238](#)
  - routeList, [1238](#)
  - RouteName, [1238](#)
  - segCount, [1238](#)
  - segmentId, [1239](#)
  - theEnd, [1239](#)
  - TheType, [1236](#)
  - theType, [1239](#)
  - TrackGraph, [1236](#)
  - TurnoutBody, [1236](#)
  - TurnoutBodyElt, [1233](#)
  - TurnoutBodyEltType, [1233](#)
- Parsers::TurnoutBodyElt::Pos, [789](#)
  - x, [789](#)
  - y, [789](#)
- Parsers::TurnoutGraphic, [1239](#)
  - maxX, [1240](#)
  - maxY, [1240](#)
  - minX, [1240](#)
  - minY, [1241](#)
  - numSegments, [1241](#)
  - segments, [1241](#)
- Parsers::TurnoutRoutelist, [1241](#)
  - numRoutelists, [1242](#)
  - routes, [1242](#)
- partline
  - FCFSupport::PDFPrinterDevice, [785](#)
  - FCFSupport::PostScriptPrinterDevice, [798](#)
- PathName
  - FCFSupport::PathName, [759](#), [760](#)
  - TTSupport::PathName, [768](#), [769](#)
- PathSeparator
  - FCFSupport::PathName, [765](#)
  - TTSupport::PathName, [773](#)
- pathname
  - FCFSupport::PathName, [766](#)
  - TTSupport::PathName, [774](#)
- pattern
  - FCFSupport::PDFFileStructures::ResourceDictionary, [850](#)
- Pause
  - FCFSupport::PauseCallback, [776](#)
- PauseCallback
  - FCFSupport::PauseCallback, [775](#)

- Peek
  - FCFSupport::Car, [252](#)
- peek
  - FCFSupport::Car, [262](#)
- PI
  - GRSupport, [101](#)
- PickCar
  - FCFSupport::SwitchListElement, [1017](#)
- pickCar
  - FCFSupport::SwitchListElement, [1018](#)
- PickCarEq
  - FCFSupport::SwitchList, [1009](#)
- PickIndex
  - FCFSupport::SwitchList, [1010](#)
- pickIndex
  - FCFSupport::SwitchList, [1012](#)
- pickLoc
  - FCFSupport::SwitchListElement, [1018](#)
- PickLocation
  - FCFSupport::SwitchListElement, [1017](#)
- PickLocationEq
  - FCFSupport::SwitchList, [1010](#)
- PickTrain
  - FCFSupport::SwitchListElement, [1017](#)
- pickTrain
  - FCFSupport::SwitchListElement, [1018](#)
- PickTrainEq
  - FCFSupport::SwitchList, [1010](#)
- pindirection
  - linuxgpio, [127](#)
- pinnotype
  - linuxgpio, [127](#)
- Plate
  - FCFSupport::Car, [252](#)
- plate
  - FCFSupport::Car, [262](#)
  - FCFSupport::Industry, [536](#)
- Poll
  - CTIAcela, [54](#)
  - cmri::CMri, [282](#)
- pollid
  - raildriver::RaildriverClient, [816](#)
- populateAliasMap
  - lcc::CANGridConnect, [210](#)
- portLCombo
  - lcc::CANGridConnectOverUSBSerial, [229](#)
- portLSpin
  - lcc::CANGridConnectOverTcp, [224](#)
  - lcc::OpenLCBOverTcp, [706](#)
- portandnidDialog
  - lcc::CANGridConnectOverUSBSerial, [229](#)
- portnidandhostDialog
  - lcc::CANGridConnectOverTcp, [224](#)
  - lcc::OpenLCBOverTcp, [706](#)
- portopenp
  - CmriSupport::CmriNode, [289](#)
- pos1
  - Parsers::BezierBodyElt, [190](#)
  - Parsers::CornuBodyElt, [347](#)
  - Parsers::TurnoutBodyElt, [1237](#)
- pos2
  - Parsers::BezierBodyElt, [191](#)
  - Parsers::CornuBodyElt, [348](#)
  - Parsers::TurnoutBodyElt, [1237](#)
- posList
  - Parsers::RouteVec, [851](#)
- position
  - Parsers::TrackGraph::CompressedNodeValues, [302](#)
- positionName
  - Parsers::RouteVec, [851](#)
- post
  - LabelComboBox, [109](#)
- PostScriptPrinterDevice
  - FCFSupport::PostScriptPrinterDevice, [793](#)
- PostScriptStandardType1FontDictionary
  - FCFSupport::PDFFileStructures::PostScript↔  
StandardType1FontDictionary, [800](#)
- PowerUpMode
  - xpressnet, [158](#)
- PoweringUp
  - xpressnet::CommandStationStatus, [297](#)
- prefix
  - FCFSupport::PDFFileStructures::PageLabel↔  
Dictionary, [740](#)
- PrevTrain
  - FCFSupport::Car, [252](#)
- prevlink
  - HTMLHelp::HTMLHelp, [507](#)
- prevtrain
  - FCFSupport::Car, [262](#)
- Print
  - FCFSupport::Train, [1214](#)
- print
  - CTCPanel::CTCPanel, [385](#)
  - FCFSupport::Train, [1219](#)
- PrintAllCarTypes
  - FCFSupport::System, [1055](#)
- PrintAllLists
  - FCFSupport::System, [1056](#)
- PrintAlpha
  - FCFSupport::System, [1056](#)
- printAlpha
  - FCFSupport::System, [1104](#)
- PrintAnalysisHeader
  - FCFSupport::System, [1056](#)
- PrintAtwice
  - FCFSupport::System, [1057](#)
- printAtwice

- FCFSupport::System, [1104](#)
- PrintCarHeading
  - FCFSupport::System, [1057](#)
- PrintCarTypesHeader
  - FCFSupport::System, [1057](#)
- PrintCarTypesSummaryHeader
  - FCFSupport::System, [1057](#)
- PrintDashedLine
  - FCFSupport::System, [1058](#)
- PrintDispatch
  - FCFSupport::System, [1058](#)
- printDispatch
  - FCFSupport::System, [1105](#)
- PrintDispatcher
  - FCFSupport::System, [1058](#)
- PrintFormFeed
  - FCFSupport::System, [1059](#)
- PrintIndustryHeader
  - FCFSupport::System, [1059](#)
- PrintList
  - FCFSupport::System, [1059](#)
- printList
  - FCFSupport::System, [1105](#)
- PrintLocCommon
  - FCFSupport::System, [1060](#)
- PrintLocOneIndustry
  - FCFSupport::System, [1060](#)
- PrintLtwice
  - FCFSupport::System, [1060](#)
- printLtwice
  - FCFSupport::System, [1105](#)
- PrintOneAnalysis
  - FCFSupport::System, [1061](#)
- PrintOneCarInfo
  - FCFSupport::System, [1061](#)
- PrintOneCarLocation
  - FCFSupport::System, [1061](#)
- PrintOneCarType
  - FCFSupport::System, [1062](#)
- PrintOneIndustry
  - FCFSupport::System, [1062](#)
- printOptions
  - TTSupport::TimeTableSystem, [1153](#)
- PrintSystemBanner
  - FCFSupport::System, [1063](#)
- PrintTrainLoc
  - FCFSupport::System, [1063](#)
- PrintTrainOrderHeader
  - FCFSupport::System, [1064](#)
- PrintTrainOrders
  - FCFSupport::System, [1064](#)
- PrintYards
  - FCFSupport::System, [1064](#)
- printYards
  - FCFSupport::System, [1105](#)
- Printem
  - FCFSupport::System, [1059](#)
- printem
  - FCFSupport::System, [1105](#)
- PrinterDevice
  - FCFSupport::PrinterDevice, [804](#)
- PrinterPageSize
  - FCFSupport::PrinterDevice, [806](#)
- printerStream
  - FCFSupport::LQ24PrinterDevice, [604](#)
  - FCFSupport::PDFPrinterDevice, [785](#)
  - FCFSupport::PostScriptPrinterDevice, [798](#)
  - FCFSupport::TextPrinterDevice, [1115](#)
- printexportfiletypes
  - lcc::ConfigurationEditor, [338](#)
- Priority
  - FCFSupport::Industry, [532](#)
- priority
  - FCFSupport::Industry, [536](#)
- procSets
  - FCFSupport::PDFFileStructures::ResourceDictionary, [850](#)
- ProcessFile
  - Parsers::ParseFile, [751](#)
- ProduceEvent
  - lcc::OpenLCBNode, [694](#)
- producer
  - FCFSupport::PDFFileStructures::Information←Directory, [541](#)
- ProducerIdentified
  - lcc::OpenLCBNode, [694](#)
- ProducerRangeIdentified
  - lcc::OpenLCBNode, [695](#)
- ProductCodeId
  - RaildriverIO, [837](#)
- ProductIdCode
  - azatrax::Azatrax, [178](#)
- ProgramMode
  - nce::NCE, [657](#)
- progress
  - mainwindow, [619](#)
- progressBar
  - splash, [927](#)
- ProgressDone
  - FCFSupport::WorkInProgressCallback, [1251](#)
- ProgressStart
  - FCFSupport::WorkInProgressCallback, [1251](#)
- ProgressUpdate
  - FCFSupport::WorkInProgressCallback, [1251](#)
- properties
  - FCFSupport::PDFFileStructures::ResourceDictionary, [850](#)
- ProtocolLabelString



- lcc::OpenLCBProtocols, 708
- ProtocolSupportRequest
  - lcc::OpenLCBNode, 695
- protocolstrings
  - lcc::OpenLCBProtocols, 709
- protocolsupport
  - lcc::OpenLCBNode, 699
- PulseOff
  - CTIAcela, 54
- PulseOn
  - CTIAcela, 54
- PulseRelays
  - azatrax::SR4, 933
- PushButton
  - CTCPanel::PushButton, 811
- pushcurrenttopic
  - HTMLHelp::HTMLHelp, 507
- Put
  - FCFSupport::LQ24PrinterDevice, 601
  - FCFSupport::PDFPrinterDevice, 781
  - FCFSupport::PostScriptPrinterDevice, 795
  - FCFSupport::PrinterDevice, 806, 807
  - FCFSupport::TextPrinterDevice, 1114
- PutLine
  - FCFSupport::LQ24PrinterDevice, 601
  - FCFSupport::PDFPrinterDevice, 782
  - FCFSupport::PostScriptPrinterDevice, 796
  - FCFSupport::PrinterDevice, 807
  - FCFSupport::TextPrinterDevice, 1114
- PutPageHeader
  - FCFSupport::PostScriptPrinterDevice, 796
- putdebug
  - lcc::ConfigMemory, 309
  - lcc::ConfigOptions, 313
  - lcc::ConfigurationEditor, 335
- Q1\_State
  - azatrax::SR4, 933
- Q1\_state
  - azatrax::SR4::status1\_union, 959
- Q2\_State
  - azatrax::SR4, 933
- Q2\_state
  - azatrax::SR4::status1\_union, 959
- Q3\_State
  - azatrax::SR4, 934
- Q3\_state
  - azatrax::SR4::status1\_union, 959
- Q4\_State
  - azatrax::SR4, 934
- Q4\_state
  - azatrax::SR4::status1\_union, 960
- Query
  - CTIAcela, 55
- QuotePDFString
  - FCFSupport::PDFFileStructures, 92
- R
  - Parsers::SegVector, 882
  - Parsers::TurnoutBodyElt, 1238
- RADIANS
  - MRRXtrkCad.tab.h, 1305
- RAMCheckError
  - xpressnet::CommandStationStatus, 298
- RAMData
  - nce, 132
- RAMData8
  - nce, 132
- RDInput
  - RaildriverIO, 838
- RESERVED\_SHIFT
  - lcc::CANHeader, 232
- RadiansToDegrees
  - GRSupport, 100
- radius
  - Parsers::BezierBodyElt, 191
  - Parsers::CornuBodyElt, 348
  - Parsers::SegVector, 883
  - Parsers::TurnoutBodyElt, 1238
- RailDriverModernDesktop
  - RaildriverIO, 838
- raildriver, 143
  - eventlist, 144
  - RaildriverEvents, 144
- raildriver::RaildriverClient, 813
  - \_poller, 815
  - \_readevent, 815
  - ~RaildriverClient, 815
  - clear, 815
  - leds, 815
  - mask, 816
  - pollid, 816
  - RaildriverClient, 815
  - socket, 816
  - speaker, 816
- RaildriverClient
  - raildriver::RaildriverClient, 815
- RaildriverClientModule, 69
- RaildriverEvents
  - raildriver, 144
- RaildriverIO, 817
  - ~RaildriverIO, 823
  - AutoBrake, 835
  - BailOff, 835
  - Digital1, 836
  - Digital2, 836
  - Digital3, 836
  - Digital4, 836

- Digital5, [836](#)
- Digital6, [836](#)
- Eventcodes, [821](#)
- Eventmask\_bits, [822](#)
- GetAlert, [824](#)
- GetAutoBrake, [824](#)
- GetBailOff, [824](#)
- GetBell, [824](#)
- GetBlueButton1, [825](#)
- GetBlueButton10, [825](#)
- GetBlueButton11, [825](#)
- GetBlueButton12, [825](#)
- GetBlueButton13, [825](#)
- GetBlueButton14, [826](#)
- GetBlueButton15, [826](#)
- GetBlueButton16, [826](#)
- GetBlueButton17, [826](#)
- GetBlueButton18, [826](#)
- GetBlueButton19, [827](#)
- GetBlueButton2, [827](#)
- GetBlueButton20, [827](#)
- GetBlueButton21, [827](#)
- GetBlueButton22, [827](#)
- GetBlueButton23, [828](#)
- GetBlueButton24, [828](#)
- GetBlueButton25, [828](#)
- GetBlueButton26, [828](#)
- GetBlueButton27, [828](#)
- GetBlueButton28, [829](#)
- GetBlueButton3, [829](#)
- GetBlueButton4, [829](#)
- GetBlueButton5, [829](#)
- GetBlueButton6, [829](#)
- GetBlueButton7, [830](#)
- GetBlueButton8, [830](#)
- GetBlueButton9, [830](#)
- GetEBrakeDown, [830](#)
- GetEBrakeUp, [830](#)
- GetHeadlight, [831](#)
- GetIndependBrake, [831](#)
- GetPanDown, [831](#)
- GetPanLeft, [831](#)
- GetPanRight, [831](#)
- GetPanUp, [832](#)
- GetPantograph, [832](#)
- GetProductCodeId, [832](#)
- GetRangeDown, [832](#)
- GetRangeUp, [832](#)
- GetReverser, [833](#)
- GetSand, [833](#)
- GetThrottle, [833](#)
- GetWhistleDown, [833](#)
- GetWhistleUp, [833](#)
- GetWiper, [834](#)
- GetZoomUp, [834](#)
- GetZoopDown, [834](#)
- Headlight, [837](#)
- IndependBrake, [837](#)
- LEDCommand, [837](#)
- PIEngineering, [837](#)
- ProductCodeId, [837](#)
- RDInput, [838](#)
- RailDriverModernDesktop, [838](#)
- RaildriverIO, [823](#)
- rdriverdev, [838](#)
- ReadInputs, [834](#)
- ReportBuffer, [838](#)
- Reverser, [839](#)
- SetLEDS, [835](#)
- SpeakerCommand, [839](#)
- SpeakerOff, [835](#)
- SpeakerOn, [835](#)
- theBytes, [839](#)
- Throttle, [839](#)
- Wiper, [839](#)
- RanAllTrains
  - FCFSupport::System, [1064](#)
- ranAllTrains
  - FCFSupport::System, [1106](#)
- Random
  - FCFSupport::System, [1065](#)
- Randomize
  - FCFSupport::System, [1065](#)
- RawPacket
  - nce, [132](#)
- RawTrackPacket
  - nce, [132](#)
- rawnode
  - Parsers::TrackGraph::CompressedNodeValues, [302](#)
- rdriverdev
  - RaildriverIO, [838](#)
- ReLoadCarFile
  - FCFSupport::System, [1068](#)
- Read
  - CTIAcela, [55](#)
  - cmri::CMri, [282](#)
  - TTSupport::Cab, [195](#)
  - TTSupport::Occupied, [681](#)
  - TTSupport::Station, [951](#)
  - TTSupport::Stop, [981](#)
  - TTSupport::StorageTrack, [988](#)
  - TTSupport::TimeRange, [1128](#)
  - TTSupport::Train, [1200](#)
- read
  - linuxgpio::LinuxGpio, [572](#)
- Read16
  - CTIAcela, [55](#)
- Read4

- CTIAcela, [56](#)
- Read8
  - CTIAcela, [56](#)
- ReadAll
  - CTIAcela, [57](#)
- ReadCVInDirectMode
  - nce::NCE, [657](#)
- ReadCVInPagedMode
  - nce::NCE, [657](#)
- ReadCarTypes
  - FCFSupport::System, [1065](#)
- ReadConfiguration, [145](#)
  - ConfigurationType, [146](#)
  - IsEven, [146](#)
  - ReadConfiguration, [147](#)
  - WriteConfiguration, [147](#)
- ReadDivisions
  - FCFSupport::System, [1066](#)
- ReadFromRAM
  - nce::NCE, [658](#)
- ReadGroupLimit
  - FCFSupport::System, [1066](#)
- ReadIndustries
  - FCFSupport::System, [1066](#)
- ReadInputs
  - RaildriverIO, [834](#)
- ReadNote
  - TTSupport::TimeTableSystem, [1148](#)
- ReadOneByteFromRAM
  - nce::NCE, [658](#)
- ReadOwners
  - FCFSupport::System, [1067](#)
- ReadRegister
  - nce::NCE, [658](#)
- ReadRevision
  - CTIAcela, [57](#)
- ReadStations
  - FCFSupport::System, [1067](#)
- ReadTrainOrders
  - FCFSupport::System, [1067](#)
- ReadTrains
  - FCFSupport::System, [1068](#)
- readevent
  - xpressnet::XPressNet, [1267](#)
- readlist
  - lcc::ConfigMemory, [310](#)
- realized
  - ScrollTabNotebook, [877](#)
- Rectangle
  - FCFSupport::PDFFileStructures::Rectangle, [841](#)
- RegisterModeRead
  - xpressnet::XPressNet, [1267](#)
- RegisterModeWrite
  - xpressnet::XPressNet, [1267](#)
- RelaysOff
  - azatrax::SR4, [934](#)
- RelaysOn
  - azatrax::SR4, [935](#)
- ReleaseTrainDisplay
  - FCFSupport::TrainDisplayCallback, [1222](#)
- Reload
  - FCFSupport::Industry, [532](#)
- reload
  - FCFSupport::Industry, [536](#)
- remLen
  - FCFSupport::Industry, [536](#)
- remoteeval
  - Satellite, [853](#)
- removeChild
  - SimpleDOMEElement, [907](#)
- RemoveLocomotiveFromMultiUnit
  - nce::NCE, [659](#)
  - xpressnet::XPressNet, [1268](#)
- RemoveNote
  - TTSupport::Stop, [981](#)
- RemoveNoteFromStop
  - TTSupport::Train, [1201](#)
- RemoveNoteFromTrain
  - TTSupport::Train, [1201](#)
- RemovedStoredTrain
  - TTSupport::StorageTrack, [989](#)
- render
  - HTMLHelp::HTMLHelp, [507](#)
- replyExpected
  - lcc::CanMessage, [237](#)
- ReportAnalysis
  - FCFSupport::System, [1068](#)
- ReportBuffer
  - RaildriverIO, [838](#)
- ReportCarLocations
  - FCFSupport::System, [1069](#)
- ReportCarOwners
  - FCFSupport::System, [1069](#)
- ReportCarTypes
  - FCFSupport::System, [1070](#)
- ReportCars
  - FCFSupport::System, [1069](#)
- ReportCarsNotMoved
  - FCFSupport::System, [1070](#)
- ReportIndustries
  - FCFSupport::System, [1071](#)
- ReportLocAll
  - FCFSupport::System, [1071](#)
- ReportLocDivision
  - FCFSupport::System, [1072](#)
- ReportLocIndustry
  - FCFSupport::System, [1072](#)
- ReportLocStation

- FCFSupport::System, [1073](#)
- ReportTrains
  - FCFSupport::System, [1073](#)
- RequestForServiceModeResults
  - xpressnet::XPressNet, [1268](#)
- requiredOpts
  - lcc::CANGridConnectOverCANSocket, [219](#)
  - lcc::CANGridConnectOverTcp, [223](#)
  - lcc::CANGridConnectOverUSBSerial, [228](#)
  - lcc::OpenLCBOverTcp, [704](#)
- reserveAlias
  - lcc::CANGridConnect, [211](#)
- reserved
  - azatrax::Azatrax::StateDataPacket, [939](#)
  - azatrax::MRD::status1\_union, [964](#)
  - azatrax::MRD::status2\_union, [966](#)
  - azatrax::SL2::status3\_union, [974](#)
  - azatrax::SR4::status2\_union, [968](#)
  - azatrax::SR4::status3\_union, [972](#)
- reserved
  - azatrax::SL2::status1\_union, [962](#)
  - azatrax::SL2::status2\_union, [970](#)
  - azatrax::SR4::status1\_union, [960](#)
- ResetIndustryStats
  - FCFSupport::System, [1073](#)
- ResetLastIndex
  - FCFSupport::SwitchList, [1011](#)
- ResetNetwork
  - CTIAcela, [57](#)
- ResetStatus
  - azatrax::MRD, [626](#)
- resetStatus
  - azatrax::MRD::status2\_union, [966](#)
- ResetStopwatch
  - azatrax::MRD, [627](#)
- ResetSwitchList
  - FCFSupport::SwitchList, [1011](#)
- resize
  - OvalWidgets::OvalScrollBar, [728](#)
- ResourceDictionary
  - FCFSupport::PDFFileStructures::ResourceDictionary, [845](#)
- resources
  - FCFSupport::PDFFileStructures::Page, [737](#)
  - FCFSupport::PDFFileStructures::PageTree, [749](#)
- responseList
  - xpressnet::XPressNet, [1272](#)
- ResponseType
  - xpressnet::CommandStationResponse, [295](#)
- Responses
  - CTIAcela, [62](#)
- RestartLoop
  - FCFSupport::System, [1074](#)
- RestoreLEDFunction
  - azatrax::Azatrax, [178](#)
- ResumeOperations
  - xpressnet::XPressNet, [1268](#)
- ReturnAuxiliaryInputUnit
  - nce::NCE, [659](#)
- ReturnAuxiliaryInputUnitShortForm
  - nce::NCE, [660](#)
- ReturnClock
  - nce::NCE, [660](#)
- ReturnMySimpleNodeInfo
  - lcc::OpenLCBNode, [695](#)
- ReturnMySupportedProtocols
  - lcc::OpenLCBNode, [696](#)
- Reverse
  - nce, [134](#)
  - xpressnet, [157](#)
- reverse
  - CabWidgets::LocomotiveDirection, [579](#)
- ReverseActionScript
  - Parsers::LayoutFile, [563](#)
  - Parsers::TrackGraph, [1191](#)
- ReverseBlink
  - CTIAcela, [57](#)
- reverseactionscrip
  - Parsers::TrackGraph::NodeValues, [677](#)
- Reverser
  - RaildriverIO, [839](#)
- right
  - ScrollTabNotebook, [877](#)
- rightbuttons
  - CabWidgets::LocomotiveSpeed, [593](#)
- rootDictionary
  - FCFSupport::PDFPrinterDevice, [786](#)
- rootnode
  - ParseXML, [758](#)
- Roots
  - Parsers::LayoutFile, [563](#)
  - Parsers::TrackGraph, [1191](#)
- RotationUnit
  - Parsers::TrackGraph, [1175](#)
- routeLength
  - Parsers::RouteVec, [851](#)
- routeList
  - Parsers::TurnoutBodyElt, [1238](#)
- RouteName
  - Parsers::TurnoutBodyElt, [1238](#)
- routes
  - Parsers::TurnoutRoutelist, [1242](#)
- RunAllTrains
  - FCFSupport::System, [1074](#)
- RunBoxMoves
  - FCFSupport::System, [1074](#)
- RunOneLocal
  - FCFSupport::System, [1075](#)

- RunOneManifest
  - FCFSupport::System, [1075](#)
- RunOnePassenger
  - FCFSupport::System, [1076](#)
- RunOneTrain
  - FCFSupport::System, [1076](#)
- S128
  - nce, [134](#)
  - xpressnet, [159](#)
- S14
  - nce, [134](#)
  - xpressnet, [159](#)
- S27
  - xpressnet, [159](#)
- S28
  - nce, [134](#)
  - xpressnet, [159](#)
- S\_128
  - xpressnet, [155](#)
- S\_14
  - xpressnet, [155](#)
- S\_27
  - xpressnet, [156](#)
- S\_28
  - xpressnet, [156](#)
- SIGPlate
  - CTCPanel::SIGPlate, [899](#)
- SIMPLE\_MASK
  - lcc::MTIDetail, [640](#)
- SIMPLE\_SHIFT
  - lcc::MTIDetail, [640](#)
- SL2
  - azatrax::Azatrax, [180](#)
  - azatrax::SL2, [915](#)
- SMile
  - TTSupport::Station, [952](#)
  - TTSupport::TimeTableSystem, [1149](#)
- SPECIAL\_MASK
  - lcc::MTIDetail, [640](#)
- SR4
  - azatrax::Azatrax, [180](#)
  - azatrax::SR4, [930](#)
- SRCID\_MASK
  - lcc::CANHeader, [232](#)
- SRCID\_SHIFT
  - lcc::CANHeader, [232](#)
- SRQControl
  - CTIAcela, [60](#)
- STREAMDG\_MASK
  - lcc::MTIDetail, [640](#)
- STX
  - cmri::CMri, [282](#)
- SWIG\_name
  - TclSocketCAN.i, [1315](#)
  - tclwiringpi.i, [1319](#)
- SWIG\_version
  - TclSocketCAN.i, [1315](#)
  - tclwiringpi.i, [1319](#)
- SWPlate
  - CTCPanel::SWPlate, [1021](#)
- SameDirectory
  - FCFSupport::PathName, [765](#)
  - TTSupport::PathName, [774](#)
- Satellite, [852](#)
  - ~Satellite, [853](#)
  - remoteeval, [853](#)
  - Satellite, [853](#)
  - socket, [854](#)
- SatelliteModule, [72](#)
- SaveCars
  - FCFSupport::System, [1077](#)
- scale
  - CTCPanel::CTCPanel, [389](#)
- ScaleClockRatio
  - nce, [133](#)
- scanEol
  - Parsers::MRRXtrkCad, [633](#)
  - YY\_MRRXtrkCad\_INHERIT, [1279](#)
- scanToEND
  - Parsers::MRRXtrkCad, [634](#)
  - YY\_MRRXtrkCad\_INHERIT, [1279](#)
- SchLabel
  - CTCPanel::SchLabel, [856](#)
- schematic
  - CTCPanel::CTCPanel, [389](#)
- schematic\_crosshair
  - CTCPanel::CTCPanel, [385](#)
- schematicYscroll
  - CTCPanel::CTCPanel, [389](#)
- ScissorCrossover
  - CTCPanel::ScissorCrossover, [860](#)
- Scripts/CMri/cmri.tcl, [1320](#)
- Scripts/CTIAcela/CTIAcela.tcl, [1328](#)
- Scripts/Common/CTCPanel2.tcl, [1321](#)
- Scripts/Common/CabWidgets.tcl, [1320](#)
- Scripts/Common/CommonTclGroup.h, [1321](#)
- Scripts/Common/HTMLHelp.tcl, [1323](#)
- Scripts/Common/ParseXML.tcl, [1326](#)
- Scripts/Common/ReadConfiguration.tcl, [1326](#)
- Scripts/Common/fileentry.tcl, [1322](#)
- Scripts/Common/gettext.tcl, [1323](#)
- Scripts/Common/labelcombobox.tcl, [1324](#)
- Scripts/Common/labelselectcolor.tcl, [1325](#)
- Scripts/Common/labelspinbox.tcl, [1325](#)
- Scripts/Common/mainwindow.tcl, [1326](#)
- Scripts/Common/panedw.tcl, [1326](#)
- Scripts/Common/snitScrollNotebook.tcl, [1327](#)

- Scripts/Common/splash.tcl, 1327
- Scripts/ControlSupport/CmriSupport.tcl, 1327
- Scripts/GRSupport/GRSupportTclGroup.h, 1329
- Scripts/GRSupport/Instruments2.tcl, 1329
- Scripts/GRSupport/LCARSWidgets2.tcl, 1330
- Scripts/GRSupport/OvalWidgets2.tcl, 1330
- Scripts/GRSupport/grsupport2.tcl, 1328
- Scripts/LCC/ConfigDialogs.tcl, 1331
- Scripts/LCC/ConfigurationEditor.tcl, 1332
- Scripts/LCC/eventDialogs.tcl, 1332
- Scripts/LCC/lcc.tcl, 1333
- Scripts/LinuxGpio/LinuxGpio.tcl, 1335
- Scripts/NCE/nce.tcl, 1336
- Scripts/RailDriverSupport/raildriver\_client.tcl, 1337
- Scripts/Satellite/Satellite.tcl, 1338
- Scripts/XPressNet/xpressnet.tcl, 1338
- scroll
  - lcc::ConfigurationEditor, 338
- ScrollTabNotebook, 862
  - \_Configure, 866
  - \_clientRow, 874
  - \_compute\_height, 866
  - \_compute\_width, 866
  - \_draw\_arrows, 867
  - \_draw\_page, 867
  - \_get\_x\_page, 867
  - \_highlight, 868
  - \_hpage, 874
  - \_left, 875
  - \_paddingtype, 875
  - \_radiustype, 875
  - \_redraw, 868
  - \_resize, 868
  - \_right, 875
  - \_select, 868
  - \_tabrow, 875
  - \_tabsides, 875
  - \_test\_page, 869
  - \_textid, 876
  - \_themeChanged, 869
  - \_themeChanged\_, 869
  - \_wpage, 876
  - \_xview, 869
  - add, 870
  - base, 876
  - compute\_size, 870
  - dbg, 876
  - forget, 871
  - get3dcolor, 871
  - index, 871
  - insert, 872
  - lbg, 876
  - left, 877
  - pages, 877
  - pages\_opts, 877
  - realized, 877
  - right, 877
  - ScrollTabNotebook, 865
  - see, 872
  - select, 873, 877
  - tab, 873
  - tabrow, 878
  - tabs, 874, 878
- scrollwindow
  - mainwindow, 620
- SearchForCarIndexesByNumber
  - FCFSupport::System, 1077
- SearchForIndustryPattern
  - FCFSupport::System, 1078
- SearchForTrainPattern
  - FCFSupport::System, 1078
- searchbackward
  - HTMLHelp::HTMLHelp, 508
- searchforward
  - HTMLHelp::HTMLHelp, 508
- see
  - ScrollTabNotebook, 872
- segCount
  - Parsers::BezierBodyElt, 191
  - Parsers::CornuBodyElt, 348
  - Parsers::TurnoutBodyElt, 1238
- segmentId
  - Parsers::BezierBodyElt, 191
  - Parsers::CornuBodyElt, 348
  - Parsers::TurnoutBodyElt, 1239
- segments
  - Parsers::TrackGraph::CompressedNodeValues, 302
  - Parsers::TurnoutGraphic, 1241
- select
  - HTMLHelp::HTMLHelp, 508
  - ScrollTabNotebook, 873, 877
- SelectLocomotive
  - CabWidgets::SelectLocomotive, 884
- selectTransportConstructor
  - lcc::OpenLCBNode, 696
- selectTransportConstructorDialog
  - lcc::OpenLCBNode, 699
- send2Bytes
  - azatrax::Azatrax, 178
- send3Bytes
  - azatrax::Azatrax, 179
- sendByte
  - azatrax::Azatrax, 179
- SendDatagram
  - lcc::OpenLCBNode, 696
- SendEvent
  - lcc::SendEvent, 887

- SendInitComplete
  - lcc::OpenLCBNode, [697](#)
- sendMessage
  - lcc::CANGridConnect, [211](#)
  - lcc::OpenLCBOverTcp, [704](#)
- SendMyNodeVerification
  - lcc::OpenLCBNode, [697](#)
- SendMySimpleNodeInfo
  - lcc::OpenLCBNode, [697](#)
- SendMySupportedProtocols
  - lcc::OpenLCBNode, [697](#)
- sendOpenLCBMessage
  - lcc::CANGridConnect, [211](#)
- SendSimpleNodeInfoRequest
  - lcc::OpenLCBNode, [698](#)
- SendSupportedProtocolsRequest
  - lcc::OpenLCBNode, [698](#)
- SendVerifyNodeID
  - lcc::OpenLCBNode, [698](#)
- sendevent
  - lcc::EventLog, [440](#)
- Sense\_1
  - azatrax::MRD, [627](#)
  - azatrax::SL2, [918](#)
- sense\_1
  - azatrax::MRD::status1\_union, [964](#)
  - azatrax::SL2::status2\_union, [970](#)
  - azatrax::SR4::status2\_union, [968](#)
- Sense\_1\_Latch
  - azatrax::SR4, [935](#)
- Sense\_1\_Live
  - azatrax::SR4, [935](#)
- Sense\_2
  - azatrax::MRD, [627](#)
  - azatrax::SL2, [918](#)
- sense\_2
  - azatrax::MRD::status1\_union, [964](#)
  - azatrax::SL2::status2\_union, [970](#)
  - azatrax::SR4::status2\_union, [968](#)
- Sense\_2\_Latch
  - azatrax::SR4, [935](#)
- Sense\_2\_Live
  - azatrax::SR4, [936](#)
- Sense\_3
  - azatrax::SL2, [919](#)
- sense\_3
  - azatrax::SL2::status2\_union, [970](#)
  - azatrax::SR4::status2\_union, [968](#)
- Sense\_3\_Latch
  - azatrax::SR4, [936](#)
- Sense\_3\_Live
  - azatrax::SR4, [936](#)
- Sense\_4
  - azatrax::SL2, [919](#)
- sense\_4
  - azatrax::SL2::status2\_union, [970](#)
  - azatrax::SR4::status2\_union, [968](#)
- Sense\_4\_Latch
  - azatrax::SR4, [936](#)
- Sense\_4\_Live
  - azatrax::SR4, [937](#)
- SenseScript
  - Parsers::LayoutFile, [563](#)
  - Parsers::TrackGraph, [1191](#)
- sensescript
  - Parsers::TrackGraph::NodeValues, [677](#)
- SerialNumber
  - azatrax::Azatrax, [180](#)
- ServiceMode
  - xpressnet::CommandStationStatus, [298](#)
  - xpressnet::ServiceModeResponse, [889](#)
- ServiceModeResponse
  - xpressnet::ServiceModeResponse, [889](#)
- SessionNumber
  - FCFSupport::System, [1078](#)
- sessionNumber
  - FCFSupport::System, [1106](#)
- Set
  - linuxgpio::GpioOutputSafeHighInvert, [460](#)
  - linuxgpio::GpioOutputSafeLowInverted, [465](#)
  - linuxgpio::LinuxGpio, [572](#)
- set
  - OvalWidgets::OvalScale, [718](#)
  - OvalWidgets::OvalSlider, [722](#)
  - OvalWidgets::OvalScrollBar, [729](#)
- SetAssignments
  - FCFSupport::Car, [253](#)
- setAttribute
  - SimpleDOMEElement, [908](#)
- SetBinaryCommandEchoMode
  - nce::NCE, [660](#)
- setByte
  - lcc::GridConnectMessage, [470](#)
- SetCab
  - TTSupport::Stop, [981](#)
- SetCabBusAddressOfUSBBoard
  - nce::NCE, [661](#)
- SetCarOwner
  - FCFSupport::Car, [253](#)
- SetChan1
  - azatrax::MRD, [627](#)
- SetChan2
  - azatrax::MRD, [628](#)
- SetClock
  - nce::NCE, [661](#)
- SetClockFormat
  - nce::NCE, [661](#)
- SetClockRatio

- nce::NCE, [662](#)
- SetCommandStationPowerUpMode
  - xpressnet::XPressNet, [1268](#)
- setData
  - lcc::CanMessage, [237](#)
  - lcc::GridConnectReply, [477](#)
- setDefault
  - HTMLHelp::HTMLHelp, [508](#)
- SetDeparture
  - TTSupport::Train, [1201](#)
- SetDestination
  - FCFSupport::Car, [253](#)
- SetDestinationStorageTrack
  - TTSupport::Train, [1201](#)
- SetDivisions
  - FCFSupport::Car, [253](#)
- SetDone
  - FCFSupport::Car, [254](#)
- SetDuplicateStationIndex
  - TTSupport::Station, [951](#)
  - TTSupport::TimeTableSystem, [1148](#)
- setElement
  - lcc::GridConnectReply, [478](#)
- SetFixedRouteP
  - FCFSupport::Car, [254](#)
- SetFunctionStateGroup1
  - xpressnet::XPressNet, [1269](#)
- SetFunctionStateGroup2
  - xpressnet::XPressNet, [1269](#)
- SetFunctionStateGroup3
  - xpressnet::XPressNet, [1270](#)
- setHeader
  - lcc::CANHeader, [231](#)
  - lcc::CanMessage, [238](#)
  - lcc::GridConnectMessage, [470](#)
  - lcc::MTIDetail, [638](#)
  - lcc::MTIHeader, [643](#)
- setHexDigit
  - lcc::GridConnectMessage, [470](#)
- SetLEDS
  - RaildriverIO, [835](#)
- SetLI101Address
  - xpressnet::XPressNet, [1270](#)
- SetLastTrain
  - FCFSupport::Car, [254](#)
- SetLayover
  - TTSupport::Stop, [982](#)
- SetLdLmt
  - FCFSupport::Car, [254](#)
- SetLength
  - FCFSupport::Car, [255](#)
- SetLocation
  - FCFSupport::Car, [255](#)
- SetLocoSpeedMode
  - nce::NCE, [664](#)
- SetLocomotiveFunctionsGroup1
  - nce::NCE, [662](#)
  - xpressnet::XPressNet, [1270](#)
- SetLocomotiveFunctionsGroup2
  - nce::NCE, [663](#)
  - xpressnet::XPressNet, [1271](#)
- SetLocomotiveFunctionsGroup3
  - nce::NCE, [663](#)
  - xpressnet::XPressNet, [1271](#)
- SetLocomotiveSpeedAndDirection
  - nce::NCE, [664](#)
  - xpressnet::XPressNet, [1271](#)
- SetLtWt
  - FCFSupport::Car, [255](#)
- SetMarks
  - FCFSupport::Car, [255](#)
- SetMaxLength
  - FCFSupport::Train, [1214](#)
- SetMaxWeight
  - FCFSupport::Train, [1215](#)
- setMessageHandler
  - lcc::CANGridConnect, [212](#)
  - lcc::OpenLCBOverTcp, [704](#)
- SetName
  - TTSupport::StorageTrack, [989](#)
- SetNotDone
  - FCFSupport::Car, [256](#)
- SetNote
  - TTSupport::TimeTableSystem, [1149](#)
- setNumDataElements
  - lcc::CanMessage, [238](#)
- SetNumber
  - FCFSupport::Car, [256](#)
- SetObjectNumber
  - FCFSupport::PDFFileStructures::IndirectObject, [518](#)
- SetOkToMirrorP
  - FCFSupport::Car, [256](#)
- SetOriginStorageTrack
  - TTSupport::Train, [1202](#)
- SetPeek
  - FCFSupport::Car, [256](#)
- SetPlate
  - FCFSupport::Car, [257](#)
- SetPrevTrain
  - FCFSupport::Car, [257](#)
- SetPrint
  - FCFSupport::Train, [1215](#)
- SetPrintAlpha
  - FCFSupport::System, [1078](#)
- SetPrintAtwice
  - FCFSupport::System, [1080](#)
- SetPrintDispatch
  - FCFSupport::System, [1080](#)



- SetPrintList
  - FCFSupport::System, 1081
- SetPrintLtwice
  - FCFSupport::System, 1081
- SetPrintOption
  - TTSupport::TimeTableSystem, 1149
- SetPrintYards
  - FCFSupport::System, 1081
- SetPrintem
  - FCFSupport::System, 1080
- SetQ1Q2open
  - azatrax::SL2, 920
- SetQ1negQ2pos
  - azatrax::SL2, 919
- SetQ1posQ2neg
  - azatrax::SL2, 919
- SetQ3Q4open
  - azatrax::SL2, 920
- SetQ3negQ4pos
  - azatrax::SL2, 920
- SetQ3posQ4neg
  - azatrax::SL2, 920
- SetShift
  - FCFSupport::Train, 1215
- SetSignalAspect
  - ncc::NCE, 665
- SetStorageTrackName
  - TTSupport::Stop, 982
- SetTransitStorageTrack
  - TTSupport::Train, 1202
- SetType
  - FCFSupport::Car, 257
- SetTypeSlant
  - FCFSupport::LQ24PrinterDevice, 601
  - FCFSupport::PDFPrinterDevice, 782
  - FCFSupport::PostScriptPrinterDevice, 796
  - FCFSupport::PrinterDevice, 808
- SetTypeSpacing
  - FCFSupport::LQ24PrinterDevice, 602
  - FCFSupport::PDFPrinterDevice, 782
  - FCFSupport::PostScriptPrinterDevice, 797
  - FCFSupport::PrinterDevice, 808
- SetTypeWeight
  - FCFSupport::LQ24PrinterDevice, 602
  - FCFSupport::PDFPrinterDevice, 783
  - FCFSupport::PostScriptPrinterDevice, 797
  - FCFSupport::PrinterDevice, 808
- SetWeightClass
  - FCFSupport::Car, 257
- setZoom
  - CTCPanel::CTCPanel, 386
- setbitfield
  - CmriSupport::CmriNode, 289
- setdata
  - SimpleDOMEElement, 908
- seti
  - CTCPanel::CTCLabel, 364
  - CTCPanel::CTCPanel, 386
  - CTCPanel::CodeButton, 293
  - CTCPanel::Crossing, 352
  - CTCPanel::Crossover, 356
  - CTCPanel::CurvedBlock, 397
  - CTCPanel::DoubleSlip, 427
  - CTCPanel::EndBumper, 432
  - CTCPanel::HiddenBlock, 482
  - CTCPanel::Lamp, 549
  - CTCPanel::PushButton, 812
  - CTCPanel::SIGPlate, 900
  - CTCPanel::SWPlate, 1022
  - CTCPanel::SchLabel, 857
  - CTCPanel::ScissorCrossover, 861
  - CTCPanel::Signal, 896
  - CTCPanel::SingleSlip, 912
  - CTCPanel::StraightBlock, 996
  - CTCPanel::StubYard, 1000
  - CTCPanel::Switch, 1004
  - CTCPanel::ThreeWaySW, 1119
  - CTCPanel::ThroughYard, 1123
  - CTCPanel::Toggle, 1158
- setport
  - CmriSupport::CmriNode, 290
- setprogress
  - mainwindow, 614
- setspeed
  - CabWidgets::LocomotiveSpeed, 591
- setstatus
  - mainwindow, 614
- settime
  - Instruments::AnalogClock, 168
  - Instruments::DigitalClock, 406
- setv
  - CTCPanel::CTCLabel, 365
  - CTCPanel::CTCPanel, 386
  - CTCPanel::CodeButton, 293
  - CTCPanel::Crossing, 352
  - CTCPanel::Crossover, 356
  - CTCPanel::CurvedBlock, 397
  - CTCPanel::DoubleSlip, 427
  - CTCPanel::EndBumper, 433
  - CTCPanel::HiddenBlock, 482
  - CTCPanel::Lamp, 549
  - CTCPanel::PushButton, 813
  - CTCPanel::SIGPlate, 901
  - CTCPanel::SWPlate, 1022
  - CTCPanel::SchLabel, 857
  - CTCPanel::ScissorCrossover, 861
  - CTCPanel::Signal, 896
  - CTCPanel::SingleSlip, 912

- CTCPanel::StraightBlock, 996
- CTCPanel::StubYard, 1000
- CTCPanel::Switch, 1004
- CTCPanel::ThreeWaySW, 1119
- CTCPanel::ThroughYard, 1123
- CTCPanel::Toggle, 1159
- setValue
  - Instruments::DialInstrument, 401
  - Instruments::DigitalInstrument, 408
  - LabelComboBox, 109
  - LabelSpinBox, 118
- shading
  - FCFSupport::PDFFileStructures::ResourceDictionary, 850
- Shift
  - FCFSupport::Train, 1216
- shift
  - FCFSupport::Train, 1219
- ShiftNumber
  - FCFSupport::System, 1082
- shiftNumber
  - FCFSupport::System, 1106
- show
  - splash, 926
- ShowBanner
  - FCFSupport::ShowBannerCallback, 891
- ShowBannerCallback
  - FCFSupport::ShowBannerCallback, 891
- ShowCarMovements
  - FCFSupport::System, 1082
- ShowCarsInDivision
  - FCFSupport::System, 1082
- ShowCarsNotMoved
  - FCFSupport::System, 1083
- ShowTrainCars
  - FCFSupport::System, 1083
- ShowTrainTotals
  - FCFSupport::System, 1084
- ShowUnassignedCars
  - FCFSupport::System, 1084
- showit
  - mainwindow, 614
- Signal
  - CTCPanel::Signal, 893
- Signal2
  - CTIAcela, 58
- Signal3
  - CTIAcela, 58
- Signal4
  - CTIAcela, 58
- SignalAspects
  - Parsers::LayoutFile, 563
  - Parsers::TrackGraph, 1191
- SignalBrightness
  - CTIAcela, 59
- SignalSettings
  - CTIAcela, 59
- SimpleDOMEElement, 902
  - \_children, 909
  - \_data, 909
  - \_formattrlist, 903
  - \_quoteXML, 904
  - addchild, 904
  - attribute, 905
  - children, 905
  - data, 905
  - display, 905
  - getElementsById, 906
  - getElementsByTagName, 906
  - getParent, 906
  - isChild, 907
  - length, 907
  - removeChild, 907
  - setAttribute, 908
  - setdata, 908
  - SimpleDOMEElement, 903
  - validate, 908
- simplenodeflags
  - lcc::CANGridConnect, 215
- simplenodeinfo
  - lcc::OpenLCBNode, 700
- SingleSlip
  - CTCPanel::SingleSlip, 911
- sixbits
  - lcc, 124
- sixteenbits
  - lcc, 124
- Size
  - FCFSupport::PDFFileStructures::IndirectObject↵  
Dictionary, 523
  - FCFSupport::PDFFileStructures::PageLabelTree, 744
- SkipCommentsGets
  - FCFSupport::System, 1084
- slideout\_add
  - mainwindow, 615
- slideout\_getframe
  - mainwindow, 615
- slideout\_hide
  - mainwindow, 615
- slideout\_isshownp
  - mainwindow, 616
- slideout\_list
  - mainwindow, 616
- slideout\_reqwidth
  - mainwindow, 616
- slideout\_show
  - mainwindow, 616

- slideouts
  - mainwindow, [620](#)
- smile
  - TTSupport::Station, [952](#)
- sock
  - lcc::OpenLCBOverTcp, [706](#)
- socket
  - lcc::CANGridConnectOverCANSocket, [219](#)
  - lcc::CANGridConnectOverTcp, [224](#)
  - raildriver::RaildriverClient, [816](#)
  - Satellite, [854](#)
- SocketCAN
  - TclSocketCANModule, [38](#)
- socketnameLEntry
  - lcc::CANGridConnectOverCANSocket, [219](#)
- socketnamenidDialog
  - lcc::CANGridConnectOverCANSocket, [220](#)
- SoftReset
  - nce::NCE, [665](#)
- SoftwareVersion
  - nce::NCE, [665](#)
  - xpressnet::LI100VersionNumbers, [567](#)
  - xpressnet::SoftwareVersion, [922](#)
- source\_file
  - Parsers::ParseFile, [753](#)
- source\_line
  - Parsers::ParseFile, [753](#)
- SourceFile
  - Parsers::ParseFile, [752](#)
- space
  - lcc::ConfigMemory, [311](#)
- speaker
  - raildriver::RaildriverClient, [816](#)
- SpeakerCommand
  - RaildriverIO, [839](#)
- SpeakerOff
  - RaildriverIO, [835](#)
- SpeakerOn
  - RaildriverIO, [835](#)
- Speed
  - TTSupport::Train, [1202](#)
  - xpressnet::DoubleHeaderInformation, [418](#)
  - xpressnet::LocomotiveInformation, [583](#)
- speed
  - CabWidgets::LocomotiveSpeed, [591](#)
  - TTSupport::Train, [1205](#)
- Speed128
  - nce, [133](#)
- Speed28
  - nce, [133](#)
- SpeedMode
  - nce, [134](#)
- SpeedStepMode
  - xpressnet::DoubleHeaderInformation, [418](#)
  - xpressnet::LocomotiveInformation, [584](#)
- SpeedStepModeCode
  - xpressnet, [159](#)
- speedtype
  - CTIAcela, [45](#)
- spl
  - yy\_MRRXtrkCad\_stype, [1281](#)
- splash, [923](#)
  - CheckColor, [925](#)
  - CheckImage, [925](#)
  - currentProgress, [927](#)
  - enableClickDestroy, [926](#)
  - header, [927](#)
  - hide, [926](#)
  - icon, [927](#)
  - image, [927](#)
  - progressBar, [927](#)
  - show, [926](#)
  - splash, [925](#)
  - status, [928](#)
  - title, [928](#)
  - update, [926](#)
- Split
  - FCFSupport::PathName, [766](#)
  - TTSupport::PathName, [774](#)
- split
  - FCFSupport::System, [1085](#)
- SquareEndOptions
  - OvalWidgets, [139](#)
- standardMethods
  - CTCPanel, [81](#)
- start
  - FCFSupport::PDFFileStructures::PageLabel↔  
Dictionary, [741](#)
- StartClock
  - nce::NCE, [665](#)
- StartMode
  - xpressnet::CommandStationStatus, [298](#)
- StartSMile
  - TTSupport::Train, [1203](#)
- startSMile
  - TTSupport::Train, [1205](#)
- state
  - CTCPanel::Crossover, [357](#)
  - CTCPanel::DoubleSlip, [428](#)
  - CTCPanel::Lamp, [550](#)
  - CTCPanel::ScissorCrossover, [862](#)
  - CTCPanel::SingleSlip, [913](#)
  - CTCPanel::Switch, [1005](#)
  - CTCPanel::ThreeWaySW, [1120](#)
- stateDataPacket
  - azatrax::Azatrax, [181](#)
- Station, [34](#)
  - FCFSupport::Station, [942](#), [943](#)

- OccupiedMap, [34](#)
- StationVector, [35](#)
- StorageTrackMap, [35](#)
- TTSupport::Station, [948](#)
- station
  - FCFSupport::Industry, [537](#)
  - FCFSupport::SwitchListElement::StationOrIndustry, [954](#)
  - FCFSupport::Train::StationOrIndustry, [955](#)
- StationIndex
  - TTSupport::Stop, [982](#)
- StationMap
  - FCFSupport, [88](#)
- StationName
  - TTSupport::TimeTableSystem, [1150](#)
- StationStop
  - FCFSupport::Train, [1216](#)
- StationTimes
  - TTSupport::StationTimes, [956](#)
- StationVector
  - FCFSupport, [88](#)
  - Station, [35](#)
- stationindex
  - TTSupport::Stop, [984](#)
- stations
  - FCFSupport::Division, [414](#)
  - FCFSupport::System, [1106](#)
  - TTSupport::TimeTableSystem, [1153](#)
- StatsFile
  - FCFSupport::System, [1085](#)
- statsFile
  - FCFSupport::System, [1106](#)
- StatsLen
  - FCFSupport::Industry, [532](#)
- statsLen
  - FCFSupport::Industry, [537](#)
- StatsPeriod
  - FCFSupport::System, [1085](#)
- statsPeriod
  - FCFSupport::System, [1107](#)
- Status
  - xpressnet::FunctionStatus, [449](#)
- status
  - HTMLHelp::HTMLHelp, [512](#)
  - mainwindow, [620](#)
  - splash, [928](#)
- status1
  - azatrax::Azatrax::StateDataPacket, [939](#)
- status2
  - azatrax::Azatrax::StateDataPacket, [939](#)
- status3
  - azatrax::Azatrax::StateDataPacket, [940](#)
- status4
  - azatrax::Azatrax::StateDataPacket, [940](#)
- statusline
  - lcc::ConfigurationEditor, [338](#)
- Stop
  - TTSupport::Stop, [978](#)
- stop
  - CabWidgets::LocomotiveSpeed, [593](#)
- StopClock
  - nce::NCE, [666](#)
- StopOperations
  - xpressnet::XPressNet, [1272](#)
- StopVector
  - Train and support classes., [36](#)
- StopI
  - TTSupport::Train, [1203](#)
- stops
  - FCFSupport::Train, [1219](#)
  - TTSupport::Train, [1205](#)
- Stopwatch
  - azatrax::MRD, [628](#)
- stopwatchFract
  - Azatrax.h, [1286](#)
- stopwatchHours
  - azatrax::Azatrax::StateDataPacket, [940](#)
- stopwatchMinutes
  - azatrax::Azatrax::StateDataPacket, [940](#)
- stopwatchSeconds
  - Azatrax.h, [1286](#)
- StopwatchTicking
  - azatrax::MRD, [628](#)
- stopwatchTicking
  - azatrax::MRD::status2\_union, [966](#)
- StorageTrack
  - TTSupport::StorageTrack, [986](#)
- StorageTrackMap
  - Station, [35](#)
- StorageTrackName
  - TTSupport::Stop, [982](#)
- storageTrackName
  - TTSupport::Stop, [984](#)
- storageTracks
  - TTSupport::Station, [953](#)
- StoreTrain
  - TTSupport::StorageTrack, [989](#)
- StraightBlock
  - CTCPanel::StraightBlock, [994](#)
- stream
  - lcc, [125](#)
- StringList
  - TTSupport, [150](#)
- StringListFromString
  - TTSupport, [151](#)
- StringListToString
  - TTSupport, [152](#)
- StringToInt

- FCFSupport::System, [1085](#)
- StringToIntRange
  - FCFSupport::System, [1086](#)
- stringVector
  - FCFSupport, [88](#)
  - TTSupport, [150](#)
- StubYard
  - CTCPanel::StubYard, [998](#)
- style
  - FCFSupport::PDFFileStructures::PageLabel↔  
Dictionary, [741](#)
- subType
  - FCFSupport::PDFFileStructures::FontDictionary, [445](#)
- subject
  - FCFSupport::PDFFileStructures::Information↔  
Directory, [541](#)
- Success
  - xpressnet, [158](#)
- sval
  - yy\_MRRXtrkCad\_stype, [1281](#)
- Switch
  - CTCPanel::Switch, [1003](#)
- SwitchList
  - FCFSupport::SwitchList, [1007](#)
- switchList
  - FCFSupport::System, [1107](#)
- SwitchListElement
  - FCFSupport::SwitchListElement, [1014](#), [1015](#)
- SwitchListElementVector
  - FCFSupport, [88](#)
- sx
  - Instruments::CabSignalLamp, [199](#)
- sy
  - Instruments::CabSignalLamp, [199](#)
- Symbol
  - FCFSupport::Division, [413](#)
- symbol
  - FCFSupport::Division, [414](#)
- System
  - FCFSupport::Car, [259](#)
  - FCFSupport::CarType, [271](#)
  - FCFSupport::Division, [413](#)
  - FCFSupport::Industry, [533](#)
  - FCFSupport::Station, [945](#)
  - FCFSupport::SwitchListElement, [1017](#)
  - FCFSupport::System, [1036](#), [1037](#)
  - FCFSupport::Train, [1216](#)
- SystemFile
  - FCFSupport::System, [1086](#)
- systemFile
  - FCFSupport::System, [1107](#)
- SystemName
  - FCFSupport::System, [1087](#)
- systemName
  - FCFSupport::System, [1107](#)
- TOCP
  - TTSupport::TimeTableSystem, [1153](#)
- TRUE
  - MRRXtrkCad.tab.h, [1305](#)
- TT\_ListToStringListString
  - TimeTableSystemTcl, [31](#)
- TT\_StringListToList
  - TimeTableSystemTcl, [31](#)
- TTSupport, [148](#)
  - doubleVector, [149](#)
  - OptionHashMap, [150](#)
  - StringList, [150](#)
  - StringListFromString, [151](#)
  - StringListToString, [152](#)
  - stringVector, [150](#)
  - TrainList, [150](#)
  - TrainStationTimes, [151](#)
  - TrainTimesAtStation, [151](#)
- TTSupport::Cab, [192](#)
  - ~Cab, [193](#)
  - Cab, [193](#)
  - Color, [194](#)
  - color, [195](#)
  - Name, [194](#)
  - name, [196](#)
  - operator=, [194](#)
  - Read, [195](#)
  - Write, [195](#)
- TTSupport::Occupied, [678](#)
  - From, [680](#)
  - from, [683](#)
  - Occupied, [680](#)
  - operator=, [681](#)
  - Read, [681](#)
  - TimeTableSystem, [682](#)
  - TrainNum, [681](#)
  - TrainNum2, [682](#)
  - trainnum, [683](#)
  - trainnum2, [683](#)
  - Until, [682](#)
  - until, [683](#)
  - Write, [682](#)
- TTSupport::PathName, [766](#)
  - ~PathName, [769](#)
  - Dirname, [769](#)
  - Extension, [769](#)
  - FullPath, [770](#)
  - operator<, [771](#)
  - operator<=, [771](#)
  - operator>, [773](#)
  - operator>=, [773](#)
  - operator+, [770](#)

- operator+=, 771
- operator=, 772
- operator==, 772
- PathName, 768, 769
- PathSeparator, 773
- pathname, 774
- SameDirectory, 774
- Split, 774
- Tail, 774
- TTSupport::Station, 946
  - ~Station, 949
  - AddStorageTrack, 949
  - DuplicateStationIndex, 949
  - duplicateStationIndex, 952
  - FindStorageTrack, 949
  - FindTrackTrainsStoredOn, 950
  - FirstStorageTrack, 950
  - LastStorageTrack, 950
  - Name, 950
  - name, 952
  - NumberOfStorageTracks, 950
  - operator=, 951
  - Read, 951
  - SMile, 952
  - SetDuplicateStationIndex, 951
  - smile, 952
  - Station, 948
  - storageTracks, 953
  - Write, 952
- TTSupport::StationTimes, 955
  - Arrival, 957
  - arrival, 958
  - Departure, 957
  - departure, 958
  - Flag, 957
  - flag, 958
  - operator=, 957
  - StationTimes, 956
- TTSupport::Stop, 975
  - ~Stop, 978
  - AddNote, 979
  - cab, 983
  - Departure, 979
  - Flag, 979
  - flag, 983
  - FlagType, 977
  - Layover, 979
  - layover, 984
  - Note, 980
  - notes, 984
  - NumberOfNotes, 980
  - operator=, 980
  - Read, 981
  - RemoveNote, 981
  - SetCab, 981
  - SetLayover, 982
  - SetStorageTrackName, 982
  - StationIndex, 982
  - stationindex, 984
  - Stop, 978
  - StorageTrackName, 982
  - storageTrackName, 984
  - TheCab, 983
  - Write, 983
- TTSupport::StorageTrack, 985
  - ~StorageTrack, 986
  - FindOccupied, 987
  - FirstOccupied, 987
  - IncludesTime, 987
  - LastOccupied, 988
  - Name, 988
  - name, 992
  - occupations, 992
  - operator=, 988
  - Read, 988
  - RemovedStoredTrain, 989
  - SetName, 989
  - StorageTrack, 986
  - StoreTrain, 989
  - UpdateStoredTrain, 990
  - UpdateStoredTrain2, 990
  - UpdateStoredTrainArrival, 990
  - UpdateStoredTrainDeparture, 991
  - UsedTimeRange, 991
  - Write, 992
- TTSupport::TimeRange, 1124
  - ContainsTime, 1126
  - From, 1126
  - from, 1129
  - operator<, 1126
  - operator<=, 1127
  - operator>, 1128
  - operator>=, 1128
  - operator=, 1127
  - operator==, 1127
  - Read, 1128
  - TimeRange, 1125
  - To, 1129
  - to, 1129
  - Write, 1129
- TTSupport::TimeTableSystem, 1130
  - ~TimeTableSystem, 1134
  - AddCab, 1134
  - AddNote, 1135
  - AddStation, 1135
  - AddStorageTrack, 1135
  - AddTrain, 1136
  - AddTrainLongVersion, 1136

- cabs, [1152](#)
- ComputeTimes, [1137](#)
- CreateLaTeXTimetable, [1137](#)
- DeleteTrain, [1138](#)
- DirectionName, [1152](#)
- DuplicateStationIndex, [1138](#)
- Filename, [1139](#)
- filepath, [1152](#)
- FindCab, [1139](#)
- FindStationByName, [1139](#)
- FindStorageTrack, [1139](#)
- FindTrainByName, [1140](#)
- FindTrainByNumber, [1140](#)
- FirstCab, [1141](#)
- FirstPrintOption, [1141](#)
- FirstTrain, [1141](#)
- GetPrintOption, [1141](#)
- lthStation, [1141](#)
- LastCab, [1142](#)
- LastPrintOption, [1142](#)
- LastTrain, [1142](#)
- MakeTimeTableGroupByClass, [1142](#)
- MakeTimeTableGroupManually, [1144](#)
- MakeTimeTableOneTable, [1144](#)
- MakeTimeTableOneTableStationsCenter, [1146](#)
- MakeTimeTableOneTableStationsLeft, [1146](#)
- Name, [1147](#)
- name, [1152](#)
- Note, [1147](#)
- notes, [1152](#)
- NumberOfCabs, [1147](#)
- NumberOfNotes, [1147](#)
- NumberOfStations, [1147](#)
- NumberOfTrains, [1148](#)
- printOptions, [1153](#)
- ReadNote, [1148](#)
- SMile, [1149](#)
- SetDuplicateStationIndex, [1148](#)
- SetNote, [1149](#)
- SetPrintOption, [1149](#)
- StationName, [1150](#)
- stations, [1153](#)
- TOCP, [1153](#)
- TimeInterval, [1150](#)
- TimeScale, [1150](#)
- TimeTableSystem, [1133](#), [1134](#)
- timeinterval, [1153](#)
- timescale, [1153](#)
- TotalLength, [1150](#)
- trains, [1153](#)
- WriteNewTimeTableFile, [1150](#)
- WriteNote, [1151](#)
- WriteOldTimeTableFile, [1151](#)
- TTSupport::Train, [1196](#)
- AddNoteToStop, [1198](#)
- AddNoteToTrain, [1198](#)
- ClassNumber, [1199](#)
- classnumber, [1204](#)
- Departure, [1199](#)
- departure, [1204](#)
- Name, [1199](#)
- name, [1204](#)
- Note, [1199](#)
- notes, [1205](#)
- Number, [1200](#)
- number, [1205](#)
- NumberOfNotes, [1200](#)
- NumberOfStops, [1200](#)
- Read, [1200](#)
- RemoveNoteFromStop, [1201](#)
- RemoveNoteFromTrain, [1201](#)
- SetDeparture, [1201](#)
- SetDestinationStorageTrack, [1201](#)
- SetOriginStorageTrack, [1202](#)
- SetTransitStorageTrack, [1202](#)
- Speed, [1202](#)
- speed, [1205](#)
- StartSMile, [1203](#)
- startSMile, [1205](#)
- StopI, [1203](#)
- stops, [1205](#)
- Train, [1197](#)
- UpdateStopCab, [1203](#)
- UpdateStopLayover, [1203](#)
- Write, [1204](#)
- TTSupport::eqstr, [433](#)
- operator(), [434](#)
- TTSupport::hash, [478](#)
- operator(), [479](#)
- TYPEWITHIN\_MASK
- lcc::MTIDetail, [641](#)
- TYPEWITHIN\_SHIFT
- lcc::MTIDetail, [641](#)
- Tab
- FCFSupport::LQ24PrinterDevice, [602](#)
- FCFSupport::PDFPrinterDevice, [783](#)
- FCFSupport::PostScriptPrinterDevice, [797](#)
- FCFSupport::PrinterDevice, [809](#)
- FCFSupport::TextPrinterDevice, [1115](#)
- tab
- ScrollTabNotebook, [873](#)
- table
- FCFSupport::PDFFileStructures::IndirectObject, [520](#)
- tabrow
- ScrollTabNotebook, [878](#)
- tabs
- ScrollTabNotebook, [874](#), [878](#)
- Tail

- FCFSupport::PathName, [766](#)
- TTSupport::PathName, [774](#)
- tb
  - yy\_MRRXtrkCad\_stype, [1282](#)
- tbb
  - yy\_MRRXtrkCad\_stype, [1282](#)
- tbbe
  - yy\_MRRXtrkCad\_stype, [1282](#)
- tbe
  - yy\_MRRXtrkCad\_stype, [1282](#)
- tcb
  - yy\_MRRXtrkCad\_stype, [1282](#)
- tcbe
  - yy\_MRRXtrkCad\_stype, [1282](#)
- Tcl\_Result
  - TimeTableSystemTcl, [33](#)
- tcl\_socketpair
  - ParserClasses, [25](#)
- TclCommon, [66](#)
- TclSocketCAN.i
  - SWIG\_name, [1315](#)
  - SWIG\_version, [1315](#)
- TclSocketCANModule, [38](#)
  - i, [39](#)
  - SocketCAN, [38](#)
  - TclSocketCAN, [39](#)
  - Tclsocketcan\_SafeInit, [39](#)
- TclSocketCAN
  - TclSocketCANModule, [39](#)
- Tclsocketcan\_SafeInit
  - TclSocketCANModule, [39](#)
- Tclwiringpi
  - tclwiringpi.i, [1319](#)
- tclwiringpi.i
  - SWIG\_name, [1319](#)
  - SWIG\_version, [1319](#)
  - Tclwiringpi, [1319](#)
- TclwiringpiModule, [40](#)
- text
  - yyltype, [1284](#)
- TextPrinterDevice
  - FCFSupport::TextPrinterDevice, [1112](#)
- textscroll
  - HTMLHelp::HTMLHelp, [512](#)
- tgType
  - Parsers::SegVector, [883](#)
- tgr
  - Parsers::TrackGraph::NodeValues, [677](#)
- theBits
  - azatrax::MRD::status1\_union, [965](#)
  - azatrax::MRD::status2\_union, [967](#)
  - azatrax::SL2::status1\_union, [962](#)
  - azatrax::SL2::status2\_union, [971](#)
  - azatrax::SL2::status3\_union, [975](#)
- azatrax::SR4::status1\_union, [960](#)
- azatrax::SR4::status2\_union, [969](#)
- azatrax::SR4::status3\_union, [973](#)
- theByte
  - azatrax::MRD::status1\_union, [965](#)
  - azatrax::MRD::status2\_union, [967](#)
  - azatrax::SL2::status1\_union, [962](#)
  - azatrax::SL2::status2\_union, [971](#)
  - azatrax::SL2::status3\_union, [975](#)
  - azatrax::SR4::status1\_union, [960](#)
  - azatrax::SR4::status2\_union, [969](#)
  - azatrax::SR4::status3\_union, [973](#)
- theBytes
  - RaildriverIO, [839](#)
- TheCab
  - TTSupport::Stop, [983](#)
- TheCar
  - FCFSupport::Industry, [533](#)
  - FCFSupport::System, [1087](#)
- TheCarGroup
  - FCFSupport::System, [1087](#)
- TheCarType
  - FCFSupport::System, [1088](#)
- TheDivision
  - FCFSupport::System, [1088](#)
- theEnd
  - Parsers::BezierBodyElt, [191](#)
  - Parsers::CornuBodyElt, [348](#)
  - Parsers::TurnoutBodyElt, [1239](#)
- TheIndustry
  - FCFSupport::Station, [945](#)
  - FCFSupport::System, [1088](#)
- theList
  - FCFSupport::SwitchList, [1012](#)
- TheOwner
  - FCFSupport::System, [1089](#)
- TheStation
  - FCFSupport::Division, [413](#)
  - FCFSupport::System, [1089](#)
- TheType
  - Parsers::BezierBodyElt, [189](#)
  - Parsers::CornuBodyElt, [346](#)
  - Parsers::TurnoutBodyElt, [1236](#)
- theType
  - Parsers::BezierBodyElt, [192](#)
  - Parsers::CornuBodyElt, [349](#)
  - Parsers::TurnoutBodyElt, [1239](#)
- ThreeWaySW
  - CTCPanel::ThreeWaySW, [1117](#)
- threebits
  - lcc, [124](#)
- Throttle
  - CTIAcela, [60](#)
  - RaildriverIO, [839](#)



- ThroughYard
  - CTCPanel::ThroughYard, [1121](#)
- TimeInterval
  - TTSupport::TimeTableSystem, [1150](#)
- TimeRange
  - TTSupport::TimeRange, [1125](#)
- TimeScale
  - TTSupport::TimeTableSystem, [1150](#)
- TimeStamp
  - xpressnet::CommandStationResponse, [295](#)
- TimeTableSystem, [26](#)
  - TTSupport::Occupied, [682](#)
  - TTSupport::TimeTableSystem, [1133](#), [1134](#)
- TimeTableSystem.h
  - USE\_UNORDERED\_MAP, [1317](#)
- TimeTableSystemTcl, [27](#)
  - ForEveryCab, [27](#)
  - ForEveryNote, [28](#)
  - ForEveryPrintOption, [28](#)
  - ForEveryStation, [29](#)
  - ForEveryTrain, [29](#)
  - NewCreateTimeTable, [30](#)
  - OldCreateTimeTable, [30](#)
  - TT\_ListToStringListString, [31](#)
  - TT\_StringListToList, [31](#)
  - Tcl\_Result, [33](#)
- timeinterval
  - TTSupport::TimeTableSystem, [1153](#)
- timescale
  - TTSupport::TimeTableSystem, [1153](#)
- timestamp
  - yytype, [1284](#)
- title
  - FCFSupport::PDFFileStructures::Information←  
Directory, [541](#)
  - FCFSupport::PDFPrinterDevice, [786](#)
  - FCFSupport::PostScriptPrinterDevice, [799](#)
  - splash, [928](#)
- tmpStatus
  - FCFSupport::Car, [263](#)
- To
  - TTSupport::TimeRange, [1129](#)
- to
  - TTSupport::TimeRange, [1129](#)
- toString
  - lcc::CanMessage, [238](#)
  - lcc::OpenLCBMessage, [686](#)
- toc
  - HTMLHelp::HTMLHelp, [512](#)
- toc\_css
  - HTMLHelp::HTMLHelp, [512](#)
- tocscroll
  - HTMLHelp::HTMLHelp, [513](#)
- Today
  - FCFSupport::System, [1089](#)
- Toggle
  - CTCPanel::Toggle, [1156](#)
- toolbar\_add
  - mainwindow, [617](#)
- toolbar\_addbutton
  - mainwindow, [617](#)
- toolbar\_buttoncget
  - mainwindow, [617](#)
- toolbar\_buttonconfigure
  - mainwindow, [618](#)
- toolbar\_hide
  - mainwindow, [618](#)
- toolbar\_setbuttonstate
  - mainwindow, [618](#)
- toolbar\_show
  - mainwindow, [619](#)
- toolbars
  - mainwindow, [620](#)
- topicstack
  - HTMLHelp::HTMLHelp, [513](#)
- TotalCars
  - FCFSupport::System, [1089](#)
- TotalLength
  - TTSupport::TimeTableSystem, [1150](#)
- totalLoads
  - FCFSupport::System, [1107](#)
- totalPickups
  - FCFSupport::System, [1108](#)
- totalRevenueTons
  - FCFSupport::System, [1108](#)
- TotalShifts
  - FCFSupport::System, [1090](#)
- totalShifts
  - FCFSupport::System, [1108](#)
- totalTons
  - FCFSupport::System, [1108](#)
- tpo
  - Parsers::TrackGraph::NodeValues, [678](#)
- tr\_rotate
  - Parsers::TrackGraph, [1191](#)
- tr\_scale
  - Parsers::TrackGraph, [1192](#)
- tr\_translate
  - Parsers::TrackGraph, [1192](#)
- TrackBody
  - Parsers::TrackBody, [1161](#)
- TrackBodyElt
  - Parsers::TrackBodyElt, [1164](#)
- TrackBodyLength
  - Parsers::TrackBody, [1162](#)
- TrackGraph
  - Parsers::BezierBody, [185](#)
  - Parsers::BezierBodyElt, [190](#)

- Parsers::CornuBody, [342](#)
  - Parsers::CornuBodyElt, [347](#)
  - Parsers::TrackBody, [1163](#)
  - Parsers::TrackBodyElt, [1166](#)
  - Parsers::TrackGraph, [1176](#)
  - Parsers::TurnoutBody, [1230](#)
  - Parsers::TurnoutBodyElt, [1236](#)
- trackGraph
  - Parsers::LayoutFile, [565](#)
- TrackLen
  - FCFSupport::Industry, [533](#)
- trackLen
  - FCFSupport::Industry, [537](#)
- TrackList
  - Parsers::LayoutFile, [564](#)
  - Parsers::TrackGraph, [1192](#)
- tracklist
  - Parsers::TrackGraph::NodeValues, [678](#)
- trackworkmethods
  - CTCPanel, [82](#)
- Train
  - FCFSupport::Train, [1209](#)
  - TTSupport::Train, [1197](#)
- Train and support classes., [36](#)
  - StopVector, [36](#)
  - TrainNumberMap, [36](#)
- TrainByIndex
  - FCFSupport::System, [1090](#)
- TrainByName
  - FCFSupport::System, [1090](#)
- TrainCarPickupCheck
  - FCFSupport::System, [1091](#)
- TrainDisplayCallback
  - FCFSupport::TrainDisplayCallback, [1220](#)
- TrainDropAllCars
  - FCFSupport::System, [1091](#)
- TrainDropOneCar
  - FCFSupport::System, [1092](#)
- trainEmpties
  - FCFSupport::System, [1108](#)
- TrainIndex
  - FCFSupport::System, [1092](#)
- trainIndex
  - FCFSupport::System, [1109](#)
- trainLastLocation
  - FCFSupport::System, [1109](#)
- trainLength
  - FCFSupport::System, [1109](#)
- TrainList
  - TTSupport, [150](#)
- trainLoads
  - FCFSupport::System, [1109](#)
- TrainLocalDrops
  - FCFSupport::System, [1093](#)
- TrainLocalOriginate
  - FCFSupport::System, [1093](#)
- TrainLocalPickups
  - FCFSupport::System, [1094](#)
- trainLongest
  - FCFSupport::System, [1109](#)
- TrainManifestDrops
  - FCFSupport::System, [1094](#)
- TrainManifestPickups
  - FCFSupport::System, [1095](#)
- TrainMap
  - FCFSupport, [88](#)
- TrainNameMap
  - FCFSupport, [89](#)
- TrainNum
  - TTSupport::Occupied, [681](#)
- TrainNum2
  - TTSupport::Occupied, [682](#)
- TrainNumberMap
  - Train and support classes., [36](#)
- TrainPickupOneCar
  - FCFSupport::System, [1095](#)
- TrainPrintConsistSummary
  - FCFSupport::System, [1096](#)
- TrainPrintFinalSummary
  - FCFSupport::System, [1096](#)
- trainPrintOK
  - FCFSupport::System, [1110](#)
- TrainPrintTown
  - FCFSupport::System, [1097](#)
- TrainStationTimes
  - TTSupport, [151](#)
- TrainTimesAtStation
  - TTSupport, [151](#)
- trainTons
  - FCFSupport::System, [1110](#)
- TrainType
  - FCFSupport::Train, [1208](#)
- trainnum
  - TTSupport::Occupied, [683](#)
- trainnum2
  - TTSupport::Occupied, [683](#)
- trains
  - FCFSupport::System, [1110](#)
  - TTSupport::TimeTableSystem, [1153](#)
- TrainsFile
  - FCFSupport::System, [1097](#)
- trainsFile
  - FCFSupport::System, [1110](#)
- Transform2D
  - Parsers::TrackGraph::Transform2D, [1224](#)
- Transmit
  - cmri::CMri, [282](#)
- transport

- lcc::CanTransport, [241](#)
- lcc::OpenLCBNode, [700](#)
- transportConstructors
  - lcc::OpenLCBNode, [699](#)
- traversePrimMST
  - Parsers::TrackGraph, [1192](#)
- trb
  - yy\_MRRXtrkCad\_type, [1282](#)
- trbe
  - yy\_MRRXtrkCad\_type, [1283](#)
- trim
  - FCFSupport::System, [1097](#)
- Trips
  - FCFSupport::Car, [258](#)
- trips
  - FCFSupport::Car, [263](#)
- ttyfd
  - CTIAcela, [62](#)
  - cmri::CMri, [283](#)
  - lcc::CANGridConnectOverUSBSerial, [229](#)
  - nce::NCE, [672](#)
  - xpressnet::XPressNet, [1272](#)
- TurnoutBody
  - Parsers::TurnoutBody, [1228](#)
  - Parsers::TurnoutBodyElt, [1236](#)
- TurnoutBody.h
  - angle, [1314](#)
  - len0, [1314](#)
  - len1, [1314](#)
- TurnoutBodyElt
  - Parsers::IntegerList, [545](#)
  - Parsers::TurnoutBody, [1230](#)
  - Parsers::TurnoutBodyElt, [1233](#)
- TurnoutBodyEltType
  - Parsers::TurnoutBodyElt, [1233](#)
- TurnoutEnds
  - Parsers::TurnoutBody, [1229](#)
- TurnoutNumber
  - Parsers::LayoutFile, [564](#)
  - Parsers::TrackGraph, [1193](#)
- TurnoutRouteCount
  - Parsers::TurnoutBody, [1229](#)
- TurnoutSegmentCount
  - Parsers::TurnoutBody, [1229](#)
- TurnoutStatus
  - xpressnet::AccessoryDecoderInformation, [165](#)
- turnoutnumber
  - Parsers::TrackGraph::NodeValues, [678](#)
- twelvebits
  - lcc, [124](#)
- twobits
  - lcc, [124](#)
- Type
  - FCFSupport::Car, [258](#)
- FCFSupport::CarType, [271](#)
- FCFSupport::Industry, [533](#)
- FCFSupport::Train, [1216](#)
- type
  - FCFSupport::Car, [263](#)
  - FCFSupport::CarType, [272](#)
  - FCFSupport::Industry, [537](#)
  - FCFSupport::PDFFileStructures::TypedDictionary, [1249](#)
  - FCFSupport::Train, [1219](#)
  - Parsers::TrackGraph::NodeValues, [678](#)
- Type1FontDictionary
  - FCFSupport::PDFFileStructures::Type1Font↔Dictionary, [1243](#), [1244](#)
- TypeCode
  - xpressnet, [159](#)
- TypeOfNode
  - Parsers::LayoutFile, [564](#)
  - Parsers::TrackGraph, [1193](#)
- TypeSlant
  - FCFSupport::PrinterDevice, [803](#)
- TypeSpacing
  - FCFSupport::PrinterDevice, [803](#)
- TypeWeight
  - FCFSupport::PrinterDevice, [803](#)
- TypedDictionary
  - FCFSupport::PDFFileStructures::TypedDictionary, [1248](#)
- u10
  - xpressnet, [156](#)
- u3
  - xpressnet, [156](#)
- u7
  - xpressnet, [156](#)
- UByte
  - nce, [133](#)
- UNEXPORT
  - linuxgpio::LinuxGpio, [573](#)
- USE\_UNORDERED\_MAP
  - TimeTableSystem.h, [1317](#)
- uatype
  - cmri, [78](#)
- ubyte
  - CTIAcela, [46](#)
  - cmri, [78](#)
  - xpressnet, [156](#)
- uint32
  - lcc, [124](#)
- UnConnectedTrackEnd
  - Parsers::TrackBodyElt, [1165](#)
- UnLoad
  - FCFSupport::Car, [258](#)
- unknown

- lcc, [125](#)
- unpost
  - LabelComboBox, [110](#)
- Until
  - TTSupport::Occupied, [682](#)
- until
  - TTSupport::Occupied, [683](#)
- up1
  - CabWidgets::LocomotiveSpeed, [593](#)
- up10
  - CabWidgets::LocomotiveSpeed, [594](#)
- update
  - splash, [926](#)
- updateAliasMap
  - lcc::CANGridConnect, [212](#)
- updateAndSyncCP
  - CTCPanel::CTCPanel, [387](#)
- updateSR
  - CTCPanel::CTCPanel, [387](#)
- UpdateStopCab
  - TTSupport::Train, [1203](#)
- UpdateStopLayover
  - TTSupport::Train, [1203](#)
- UpdateStoredTrain
  - TTSupport::StorageTrack, [990](#)
- UpdateStoredTrain2
  - TTSupport::StorageTrack, [990](#)
- UpdateStoredTrainArrival
  - TTSupport::StorageTrack, [990](#)
- UpdateStoredTrainDeparture
  - TTSupport::StorageTrack, [991](#)
- UpdateTrainDisplay
  - FCFSupport::TrainDisplayCallback, [1222](#)
- Upper
  - xpressnet, [158](#)
- UpperCase
  - FCFSupport::System, [1098](#)
- Url
  - HTMLHelp::HTMLHelp, [513](#)
- usedLen
  - FCFSupport::Industry, [537](#)
- UsedTimeRange
  - TTSupport::StorageTrack, [991](#)
- VALUEFMT
  - linuxgpio::LinuxGpio, [573](#)
- VARIABLEFIELD\_MASK
  - lcc::CANHeader, [232](#)
- VARIABLEFIELD\_SHIFT
  - lcc::CANHeader, [232](#)
- VBar
  - OvalWidgets, [141](#)
- valid
  - lcc, [125](#)
- valid\_heads
  - Parsers::TrackGraph, [1195](#)
- validate
  - CTIAcela, [60](#)
  - CmriSupport::CmriNode, [290](#)
  - lcc::CanAlias, [202](#)
  - lcc::CanMessage, [238](#)
  - lcc::EventID\_or\_null, [437](#)
  - lcc::EventID, [436](#)
  - lcc::OpenLCBMessage, [686](#)
  - lcc::OpenLCBProtocols, [708](#)
  - lcc::nid\_or\_null, [673](#)
  - ParseXML, [757](#)
  - SimpleDOMEElement, [908](#)
- ValueRange
  - Instruments::DialInstrument, [402](#)
- verifyBoolMethod
  - CTCPanel, [82](#)
- VerifyBooleanMethod
  - GRSupport, [101](#)
- VerifyColorMethod
  - GRSupport, [101](#)
- verifyColorMethod
  - CTCPanel, [82](#)
- VerifyDoubleMethod
  - GRSupport, [101](#)
- verifyDoubleMethod
  - CTCPanel, [82](#)
- VerifyIntegerMethod
  - GRSupport, [101](#)
- verifyOrientation8Method
  - CTCPanel, [82](#)
- VerifyOrientationHVMMethod
  - GRSupport, [101](#)
- verifyPositionMethod
  - CTCPanel, [82](#)
- wayFreight
  - FCFSupport::System, [1110](#)
- WeightClass
  - FCFSupport::Car, [258](#)
- weightclass
  - FCFSupport::Car, [263](#)
  - FCFSupport::Industry, [538](#)
- whitespace
  - FCFSupport::System, [1111](#)
- widths
  - FCFSupport::PDFFileStructures::Type1Font↔  
Dictionary, [1246](#)
- Wiper
  - RaildriverIO, [839](#)
- wipmessage
  - mainwindow, [620](#)
- WorkInProgressCallback

- FCFSupport::WorkInProgressCallback, [1250](#)
- Write
  - TTSupport::Cab, [195](#)
  - TTSupport::Occupied, [682](#)
  - TTSupport::Station, [952](#)
  - TTSupport::Stop, [983](#)
  - TTSupport::StorageTrack, [992](#)
  - TTSupport::TimeRange, [1129](#)
  - TTSupport::Train, [1204](#)
- write
  - linuxgpio::LinuxGpio, [572](#)
- Write4BytesToRAM
  - nce::NCE, [666](#)
- Write8BytesToRAM
  - nce::NCE, [666](#)
- WriteCVInDirectMode
  - nce::NCE, [667](#)
- WriteCVInPagedMode
  - nce::NCE, [667](#)
- WriteConfiguration
  - ReadConfiguration, [147](#)
- WriteDictionaryElements
  - FCFSupport::PDFFileStructures::CatalogDictionary, [275](#)
  - FCFSupport::PDFFileStructures::Dictionary, [404](#)
  - FCFSupport::PDFFileStructures::FontDictionary, [444](#)
  - FCFSupport::PDFFileStructures::IndirectObject↔Dictionary, [523](#)
  - FCFSupport::PDFFileStructures::Information↔Directory, [540](#)
  - FCFSupport::PDFFileStructures::Page, [736](#)
  - FCFSupport::PDFFileStructures::PageLabel↔Dictionary, [740](#)
  - FCFSupport::PDFFileStructures::PageLabelTree, [744](#)
  - FCFSupport::PDFFileStructures::PageTree, [748](#)
  - FCFSupport::PDFFileStructures::ResourceDictionary, [849](#)
  - FCFSupport::PDFFileStructures::Type1Font↔Dictionary, [1245](#)
  - FCFSupport::PDFFileStructures::TypedDictionary, [1248](#)
- WriteDictionaryType
  - FCFSupport::PDFFileStructures::TypedDictionary, [1249](#)
- WriteDirect
  - FCFSupport::PDFFileStructures::Dictionary, [404](#)
  - FCFSupport::PDFFileStructures::FreedObject, [447](#)
  - FCFSupport::PDFFileStructures::IndirectFloatVector, [515](#)
  - FCFSupport::PDFFileStructures::IndirectObject, [518](#)
  - FCFSupport::PDFFileStructures::PDFStream, [787](#)
  - FCFSupport::PDFFileStructures::Rectangle, [842](#)
- WriteFontType
  - FCFSupport::PDFFileStructures::FontDictionary, [445](#)
- WriteIndirectReference
  - FCFSupport::PDFFileStructures::IndirectObject, [519](#)
- WriteLCDLine3
  - nce::NCE, [668](#)
- WriteLCDLine4
  - nce::NCE, [668](#)
- WriteLCDRightLine2
  - nce::NCE, [669](#)
- WriteNewTimeTableFile
  - TTSupport::TimeTableSystem, [1150](#)
- WriteNote
  - TTSupport::TimeTableSystem, [1151](#)
- WriteObjectToFile
  - FCFSupport::PDFFileStructures::IndirectObject, [519](#)
- WriteOldTimeTableFile
  - TTSupport::TimeTableSystem, [1151](#)
- WriteOneByteToRAM
  - nce::NCE, [669](#)
- WriteOneCarToDisk
  - FCFSupport::System, [1098](#)
- WriteRAWPacket
  - nce::NCE, [669](#)
- WriteRAWTrackPacket
  - nce::NCE, [670](#)
- WriteRegister
  - nce::NCE, [670](#)
- writeReplyCheck
  - lcc::ConfigMemory, [311](#)
  - lcc::ConfigurationEditor, [338](#)
- WriteTable
  - FCFSupport::PDFFileStructures::CrossReference↔Table, [360](#)
- WriteToRAM
  - nce::NCE, [671](#)
- WriteTwoBytesToRAM
  - nce::NCE, [671](#)
- writelengths
  - lcc::ConfigOptions, [314](#)
- writelists
  - lcc::ConfigMemory, [311](#)
- x
  - Parsers::BezierBodyElt::Pos, [790](#)
  - Parsers::CornuBodyElt::Pos, [791](#)
  - Parsers::SegPos, [879](#)
  - Parsers::TrackBodyElt, [1166](#)
  - Parsers::TrackGraph::EdgeValues, [429](#)
  - Parsers::TrackGraph::Point, [788](#)
  - Parsers::TurnoutBodyElt::Pos, [789](#)
- X1
  - FCFSupport::PDFFileStructures::Rectangle, [842](#)
- x1
  - FCFSupport::PDFFileStructures::Rectangle, [843](#)

- X2
  - FCFSupport::PDFFileStructures::Rectangle, [842](#)
- x2
  - FCFSupport::PDFFileStructures::Rectangle, [843](#)
- xObject
  - FCFSupport::PDFFileStructures::ResourceDictionary, [850](#)
- XPressNet
  - xpressnet::XPressNet, [1255](#)
- XPressNetModule, [64](#)
- XYWH
  - OvalWidgets, [140](#)
- XpressNetEvent
  - xpressnet::XpressNetEvent, [1273](#)
- xpressnet, [152](#)
  - Automatic, [158](#)
  - ConsistAddress, [155](#)
  - DecoderLongAddress, [155](#)
  - DirectionCode, [157](#)
  - ElementAddress, [155](#)
  - ErrorTypeCode, [157](#)
  - Forward, [157](#)
  - Lower, [158](#)
  - Manual, [158](#)
  - MessageTypeCode, [157](#)
  - nibble, [155](#)
  - NibbleCode, [158](#)
  - Other, [158](#)
  - PowerUpMode, [158](#)
  - Reverse, [157](#)
  - S128, [159](#)
  - S14, [159](#)
  - S27, [159](#)
  - S28, [159](#)
  - S\_128, [155](#)
  - S\_14, [155](#)
  - S\_27, [156](#)
  - S\_28, [156](#)
  - SpeedStepModeCode, [159](#)
  - Success, [158](#)
  - TypeCode, [159](#)
  - u10, [156](#)
  - u3, [156](#)
  - u7, [156](#)
  - ubyte, [156](#)
  - Upper, [158](#)
  - xpressnet::XpressNetEvent, [1275](#)
- xpressnet::AccessoryDecoderInformation, [161](#)
  - \_accessory\_type, [165](#)
  - \_address, [165](#)
  - \_completed, [165](#)
  - \_nibble, [166](#)
  - \_numberOfFeedbackElements, [166](#)
  - \_t1, [166](#)
  - \_t2, [166](#)
  - AccessoryDecoderInformation, [162](#)
  - AccessoryType, [162](#)
  - Address, [164](#)
  - Completed, [164](#)
  - Nibble, [164](#)
  - NumberOfFeedbackElements, [164](#)
  - TurnoutStatus, [165](#)
- xpressnet::CommandStationResponse, [294](#)
  - \_time\_stamp, [295](#)
  - CommandStationResponse, [295](#)
  - ResponseType, [295](#)
  - TimeStamp, [295](#)
- xpressnet::CommandStationStatus, [296](#)
  - \_RAM\_check\_error, [299](#)
  - \_emergency\_off, [298](#)
  - \_emergency\_stop, [298](#)
  - \_poweringup, [299](#)
  - \_service\_mode, [299](#)
  - \_start\_mode, [299](#)
  - CommandStationStatus, [297](#)
  - EmergencyOff, [297](#)
  - EmergencyStop, [297](#)
  - PoweringUp, [297](#)
  - RAMCheckError, [298](#)
  - ServiceMode, [298](#)
  - StartMode, [298](#)
- xpressnet::DoubleHeaderInformation, [415](#)
  - \_address, [419](#)
  - \_address2, [419](#)
  - \_available, [419](#)
  - \_direction, [419](#)
  - \_function0, [419](#)
  - \_function1, [420](#)
  - \_function10, [420](#)
  - \_function11, [420](#)
  - \_function12, [420](#)
  - \_function2, [420](#)
  - \_function3, [420](#)
  - \_function4, [421](#)
  - \_function5, [421](#)
  - \_function6, [421](#)
  - \_function7, [421](#)
  - \_function8, [421](#)
  - \_function9, [421](#)
  - \_speed, [422](#)
  - \_speedstep, [422](#)
  - Address, [417](#)
  - Address2, [417](#)
  - Available, [418](#)
  - Direction, [418](#)
  - DoubleHeaderInformation, [416](#)
  - Function, [418](#)
  - Speed, [418](#)

- SpeedStepMode, 418
- xpressnet::DoubleHeaderMuError, 422
  - \_error, 423
  - DoubleHeaderMuError, 423
  - Error, 423
- xpressnet::FunctionStatus, 447
  - \_status0, 450
  - \_status1, 450
  - \_status10, 450
  - \_status11, 450
  - \_status12, 450
  - \_status2, 451
  - \_status3, 451
  - \_status4, 451
  - \_status5, 451
  - \_status6, 451
  - \_status7, 451
  - \_status8, 452
  - \_status9, 452
  - FunctionStatus, 449
  - Status, 449
- xpressnet::LI100Message, 565
  - \_message\_type, 566
  - LI100Message, 565
  - MessageType, 566
- xpressnet::LI100VersionNumbers, 566
  - \_hardware\_version, 568
  - \_software\_version, 568
  - HardwareVersion, 567
  - LI100VersionNumbers, 567
  - SoftwareVersion, 567
- xpressnet::LI101XPressNetAddress, 568
  - \_address, 569
  - Address, 569
  - LI101XPressNetAddress, 569
- xpressnet::LocomotiveAddress, 573
  - \_address, 575
  - \_addressType, 575
  - Address, 575
  - AddressType, 575
  - LocomotiveAddress, 574
- xpressnet::LocomotiveInformation, 579
  - \_address, 584
  - \_address2, 584
  - \_available, 584
  - \_direction, 584
  - \_function0, 585
  - \_function1, 585
  - \_function10, 585
  - \_function11, 585
  - \_function12, 585
  - \_function2, 585
  - \_function3, 586
  - \_function4, 586
  - \_function5, 586
  - \_function6, 586
  - \_function7, 586
  - \_function8, 586
  - \_function9, 587
  - \_mtraddress, 587
  - \_speed, 587
  - \_speedstep, 587
  - Address, 582
  - Address2, 582
  - Available, 583
  - Direction, 583
  - Function, 583
  - LocomotiveInformation, 581
  - MTR, 583
  - Speed, 583
  - SpeedStepMode, 584
- xpressnet::ServiceModeResponse, 888
  - \_cv, 890
  - \_data, 890
  - \_service\_mode, 890
  - CV, 889
  - Data, 889
  - ServiceMode, 889
  - ServiceModeResponse, 889
- xpressnet::SoftwareVersion, 921
  - \_command\_station\_type, 923
  - \_major, 923
  - \_minor, 923
  - CommandStationTypeCode, 922
  - Major, 922
  - Minor, 922
  - SoftwareVersion, 922
- xpressnet::XPressNet, 1252
  - \_ADDRESS, 1256
  - \_CheckForResponse\_0x00, 1256
  - \_CheckForResponse\_0x40, 1257
  - \_CheckForResponse\_0x60, 1257
  - \_CheckForResponse\_0x80, 1257
  - \_CheckForResponse\_0xa0, 1257
  - \_CheckForResponse\_0xc0, 1257
  - \_CheckForResponse\_0xe0, 1258
  - \_appendXORByte, 1256
  - \_readbyte, 1258
  - \_readevent, 1258
  - \_timeout, 1272
  - \_timeoutevent, 1258
  - \_transmit, 1259
  - ~XPressNet, 1256
  - AccessoryDecoderInformationRequest, 1259
  - AccessoryDecoderOperation, 1259
  - AddLocomotiveToMultiUnit, 1260
  - AddressInquiryNextMUMember, 1260
  - AddressInquiryNextMU, 1260



- AddressInquiryNextStack, [1261](#)
- AddressInquiryPreviousMUMember, [1261](#)
- AddressInquiryPreviousMU, [1261](#)
- AddressInquiryPreviousStack, [1261](#)
- CheckForResponse, [1262](#)
- CommandStationSoftwareVersion, [1262](#)
- CommandStationStatusRequest, [1262](#)
- DeleteLocomotiveFromStack, [1262](#)
- DirectModeCVRead, [1263](#)
- DirectModeCVWrite, [1263](#)
- DissolveDoubleHeader, [1263](#)
- EmergencyStopALocomotive, [1264](#)
- EmergencyStopAllLocomotives, [1264](#)
- EstablishDoubleHeader, [1264](#)
- FunctionStatusRequest, [1264](#)
- GetLI100VersionNumbers, [1265](#)
- GetNextCommandStationResponse, [1265](#)
- LocomotiveInformationRequest, [1265](#)
- OperatingModeProgrammingBitModeWrite, [1265](#)
- OperatingModeProgrammingByteModeWrite, [1266](#)
- PagedModeCVRead, [1266](#)
- PagedModeCVWrite, [1266](#)
- readevent, [1267](#)
- RegisterModeRead, [1267](#)
- RegisterModeWrite, [1267](#)
- RemoveLocomotiveFromMultiUnit, [1268](#)
- RequestForServiceModeResults, [1268](#)
- responseList, [1272](#)
- ResumeOperations, [1268](#)
- SetCommandStationPowerUpMode, [1268](#)
- SetFunctionStateGroup1, [1269](#)
- SetFunctionStateGroup2, [1269](#)
- SetFunctionStateGroup3, [1270](#)
- SetLI101Address, [1270](#)
- SetLocomotiveFunctionsGroup1, [1270](#)
- SetLocomotiveFunctionsGroup2, [1271](#)
- SetLocomotiveFunctionsGroup3, [1271](#)
- SetLocomotiveSpeedAndDirection, [1271](#)
- StopOperations, [1272](#)
- ttyfd, [1272](#)
- XPressNet, [1255](#)
- xpressnet::XpressNetEvent, [1273](#)
  - \_eventhandler, [1274](#)
  - \_script, [1274](#)
  - ~XpressNetEvent, [1274](#)
  - XpressNetEvent, [1273](#)
  - xpressnet, [1275](#)
- xscroll
  - CTCPanel::CTCPanel, [390](#)
- y
  - Parsers::BezierBodyElt::Pos, [790](#)
  - Parsers::CornuBodyElt::Pos, [791](#)
  - Parsers::SegPos, [879](#)
  - Parsers::TrackBodyElt, [1167](#)
  - Parsers::TrackGraph::EdgeValues, [430](#)
  - Parsers::TrackGraph::Point, [788](#)
  - Parsers::TurnoutBodyElt::Pos, [789](#)
  - Y1
    - FCFSupport::PDFFileStructures::Rectangle, [842](#)
  - y1
    - FCFSupport::PDFFileStructures::Rectangle, [843](#)
  - Y2
    - FCFSupport::PDFFileStructures::Rectangle, [842](#)
  - y2
    - FCFSupport::PDFFileStructures::Rectangle, [843](#)
  - YARDSperMM
    - MRRXtrkCad.tab.h, [1311](#)
  - YY\_MRRXtrkCad\_CHAR
    - MRRXtrkCad.tab.h, [1305](#)
    - YY\_MRRXtrkCad\_INHERIT, [1280](#)
  - YY\_MRRXtrkCad\_CLASS
    - MRRXtrkCad.tab.h, [1305](#)
    - YY\_MRRXtrkCad\_INHERIT, [1278](#)
  - YY\_MRRXtrkCad\_COMPATIBILITY
    - MRRXtrkCad.tab.h, [1305](#)
  - YY\_MRRXtrkCad\_CONSTRUCTOR\_CODE
    - MRRXtrkCad.tab.h, [1306](#)
  - YY\_MRRXtrkCad\_CONSTRUCTOR\_INIT
    - MRRXtrkCad.tab.h, [1306](#)
  - YY\_MRRXtrkCad\_CONSTRUCTOR\_PARAM
    - MRRXtrkCad.tab.h, [1306](#)
  - YY\_MRRXtrkCad\_DEBUG\_FLAG
    - MRRXtrkCad.tab.h, [1306](#)
    - YY\_MRRXtrkCad\_INHERIT, [1280](#)
  - YY\_MRRXtrkCad\_DEBUG
    - MRRXtrkCad.tab.h, [1306](#)
  - YY\_MRRXtrkCad\_ENUM\_TOKEN
    - Parsers::MRRXtrkCad, [630](#)
    - YY\_MRRXtrkCad\_INHERIT, [1276](#)
  - YY\_MRRXtrkCad\_ERROR\_BODY
    - MRRXtrkCad.tab.h, [1307](#)
  - YY\_MRRXtrkCad\_ERROR\_VERBOSE
    - MRRXtrkCad.tab.h, [1307](#)
  - YY\_MRRXtrkCad\_ERROR
    - MRRXtrkCad.tab.h, [1306](#)
    - YY\_MRRXtrkCad\_INHERIT, [1278](#)
  - YY\_MRRXtrkCad\_INHERIT, [1275](#)
    - ~MRRXtrkCad, [1278](#)
    - CurrentScale, [1279](#)
    - fieldflag, [1279](#)
    - lookup\_word, [1278](#)
    - MRRXtrkCad.tab.h, [1307](#)
    - scanEol, [1279](#)
    - scanToEND, [1279](#)
    - YY\_MRRXtrkCad\_CHAR, [1280](#)
    - YY\_MRRXtrkCad\_CLASS, [1278](#)
    - YY\_MRRXtrkCad\_DEBUG\_FLAG, [1280](#)



- YY\_MRRXtrkCad\_ENUM\_TOKEN, [1276](#)
- YY\_MRRXtrkCad\_ERROR, [1278](#)
- YY\_MRRXtrkCad\_LEX, [1278](#)
- YY\_MRRXtrkCad\_LLOC, [1280](#)
- YY\_MRRXtrkCad\_LVAL, [1280](#)
- YY\_MRRXtrkCad\_NERRS, [1280](#)
- YY\_MRRXtrkCad\_PARSE, [1279](#)
- yyerror1, [1279](#)
- YY\_MRRXtrkCad\_LEX\_BODY
  - MRRXtrkCad.tab.h, [1307](#)
- YY\_MRRXtrkCad\_LEX
  - MRRXtrkCad.tab.h, [1307](#)
  - YY\_MRRXtrkCad\_INHERIT, [1278](#)
- YY\_MRRXtrkCad\_LLOC
  - MRRXtrkCad.tab.h, [1307](#)
  - YY\_MRRXtrkCad\_INHERIT, [1280](#)
- YY\_MRRXtrkCad\_LSP\_NEEDED
  - MRRXtrkCad.tab.h, [1307](#)
- YY\_MRRXtrkCad\_LTYPE
  - MRRXtrkCad.tab.h, [1308](#)
- YY\_MRRXtrkCad\_LVAL
  - MRRXtrkCad.tab.h, [1308](#)
  - YY\_MRRXtrkCad\_INHERIT, [1280](#)
- YY\_MRRXtrkCad\_MEMBERS
  - MRRXtrkCad.tab.h, [1308](#)
- YY\_MRRXtrkCad\_NERRS
  - MRRXtrkCad.tab.h, [1308](#)
  - YY\_MRRXtrkCad\_INHERIT, [1280](#)
- YY\_MRRXtrkCad\_PARSE\_PARAM
  - MRRXtrkCad.tab.h, [1308](#)
- YY\_MRRXtrkCad\_PARSE
  - MRRXtrkCad.tab.h, [1308](#)
  - YY\_MRRXtrkCad\_INHERIT, [1279](#)
- YY\_MRRXtrkCad\_STYPE
  - MRRXtrkCad.tab.h, [1309](#)
- YY\_MRRXtrkCad\_USE\_CONST\_TOKEN
  - MRRXtrkCad.tab.h, [1309](#)
- YY\_MRRXtrkCad\_USE\_GOTO
  - MRRXtrkCad.tab.h, [1309](#)
- YY\_USE\_CLASS
  - MRRXtrkCad.tab.h, [1309](#)
- yy\_MRRXtrkCad\_stype, [1280](#)
  - fval, [1281](#)
  - il, [1281](#)
  - ival, [1281](#)
  - spl, [1281](#)
  - sval, [1281](#)
  - tb, [1282](#)
  - tbb, [1282](#)
  - tbbe, [1282](#)
  - tbe, [1282](#)
  - tcb, [1282](#)
  - tcbe, [1282](#)
  - trb, [1282](#)
  - trbe, [1283](#)
- yychar
  - Parsers::MRRXtrkCad, [634](#)
- yydebug
  - Parsers::MRRXtrkCad, [634](#)
- yyerror
  - Parsers::MRRXtrkCad, [632](#)
- yyerror1
  - Parsers::MRRXtrkCad, [632](#)
  - YY\_MRRXtrkCad\_INHERIT, [1279](#)
- yylex
  - Parsers::MRRXtrkCad, [633](#)
- yyloc
  - Parsers::MRRXtrkCad, [634](#)
- yylype, [1283](#)
  - first\_column, [1283](#)
  - first\_line, [1283](#)
  - last\_column, [1283](#)
  - last\_line, [1284](#)
  - MRRXtrkCad.tab.h, [1309](#)
  - text, [1284](#)
  - timestamp, [1284](#)
- yylyal
  - Parsers::MRRXtrkCad, [634](#)
- yyneres
  - Parsers::MRRXtrkCad, [634](#)
- yyparse
  - Parsers::MRRXtrkCad, [633](#)
- zoomBy
  - CTCPanel::CTCPanel, [388](#)