Role Playing Database (API Documentation)

3.1.3

Generated by Doxygen 1.8.13

Contents

1 GNU GENERAL PUBLIC LICENSE **1**

2 Module Index **7**

2.1 Modules 7

3 Namespace Index **9**

3.1 Namespace List 9

4 Class Index **11**

4.1 Class List 11

5 Module Documentation **13**

5.1 TclCommon 13

5.1.1 Detailed Description 13

5.1.2 Function Documentation 13

5.1.2.1 DynamicHelp_include() 13

5.2 RPGSupport 14

5.2.1 Detailed Description 14

6 Namespace Documentation **15**

6.1 gettext Namespace Reference . . . 15

6.1.1 Detailed Description . . . 15

6.1.2 Package provided . . . 15

6.1.3 Function Documentation . . . 15

6.1.3.1 _() . . . 15

6.1.3.2 _m() . . . 16

6.1.3.3 _mx() . . . 16

6.2 ReadConfiguration Namespace Reference . . . 16

6.2.1 Detailed Description . . . 16

6.2.2 Function Documentation . . . 16

6.2.2.1 ConfigurationType() . . . 16

6.2.2.2 IsEven() . . . 17

6.2.2.3 ReadConfiguration() . . . 17

6.2.2.4 WriteConfiguration() . . . 18

6.3 RolePlayingDB3 Namespace Reference . . . 18

6.3.1 Detailed Description . . . 19

6.3.2 Function Documentation . . . 19

6.3.2.1 GeneratePrintDialog() . . . 19

6.3.2.2 LevelDirOption() . . . 19

6.3.2.3 LevelEditorOption() . . . 19

6.3.2.4 MapBundleMountPoint() . . . 20

6.3.2.5 MapEditorOption() . . . 20

6.3.2.6 OpenFilename() . . . 20

6.3.2.7 SaveFilename() . . . 20

6.3.2.8 SpaceFileOption() . . . 20

6.4 vfs::rpg Namespace Reference . . . 21

6.4.1 Detailed Description . . . 21

6.4.2 Function Documentation . . . 21

6.4.2.1 fail() . . . 21

6.4.2.2 Mount() . . . 21

7 Class Documentation **23**

7.1 RolePlayingDB3::chroot_chooseDirectory Class Reference . . . 23

7.1.1 Detailed Description . . . 23

7.2 RolePlayingDB3::chroot_getFile Class Reference . . . 23

7.2.1 Detailed Description . . . 23

7.3 Dialog Class Reference . . . 23

7.3.1 Detailed Description . . . 23

7.4 DynamicHelp Class Reference . . . 24

7.4.1 Detailed Description . . . 24

7.4.2 Member Function Documentation . . . 24

7.4.2.1 add() . . . 24

7.4.2.2 configure() . . . 25

7.4.2.3 delete() . . . 25

7.4.2.4 register() . . . 25

7.5 RolePlayingDB3::Graphic Class Reference . . . 26

7.5.1 Detailed Description . . . 26

7.5.2 Constructor & Destructor Documentation . . . 26

7.5.2.1 Graphic() . . . 26

7.5.3 Member Function Documentation . . . 26

7.5.3.1 getimage() . . . 27

7.6 IconBitmap Class Reference . . . 27

7.6.1 Detailed Description . . . 27

7.6.2 Member Function Documentation . . . 27

7.6.2.1 bitmap() . . . 27

7.7 IconImage Class Reference . . . 28

7.7.1 Detailed Description . . . 28

7.7.2 Member Function Documentation . . . 28

7.7.2.1 image() . . . 28

7.8 RolePlayingDB3::LabeledDirTree Class Reference 28
7.8.1 Detailed Description 29
7.8.2 Constructor & Destructor Documentation 29
7.8.2.1 LabeledDirTree() 29
7.8.3 Member Function Documentation 30
7.8.3.1 binditem() 30
7.8.3.2 itemcget() 30
7.8.3.3 redrawdirtree() 31
7.9 RolePlayingDB3::LabeledScrolledText Class Reference 31
7.9.1 Detailed Description 31
7.9.2 Constructor & Destructor Documentation 31
7.9.2.1 LabeledScrolledText() 31
7.9.3 Member Function Documentation 32
7.9.3.1 bind() 32
7.10 RolePlayingDB3::LevelEditor Class Reference 32
7.10.1 Detailed Description 33
7.10.2 Constructor & Destructor Documentation 33
7.10.2.1 LevelEditor() 33
7.10.3 Member Function Documentation 33
7.10.3.1 checksave() 33
7.10.3.2 close() 34
7.10.3.3 drawonespace() 34
7.10.3.4 drawspace() 34
7.10.3.5 new() [1/2] 35
7.10.3.6 new() [2/2] 35
7.10.3.7 open() [1/2] 35
7.10.3.8 open() [2/2] 36
7.10.3.9 print() 36

7.10.3.10 printLevel() . . . 36
7.10.3.11 recreateXML() . . . 37
7.10.3.12 removeeditor() . . . 37
7.10.3.13 rescalelevelmap() . . . 37
7.10.3.14 save() . . . 37
7.10.3.15 saveas() . . . 38
7.10.3.16 setspaceposition() . . . 38
7.10.3.17 updatelevelmap() . . . 38
7.10.3.18 updatelevelmapcolor() . . . 38
7.10.3.19 updateleveltree() . . . 39
7.10.3.20 updatemediatree() [1/2] . . . 39
7.10.3.21 updatemediatree() [2/2] . . . 39
7.10.3.22 updatespacetree() . . . 39
7.11 RolePlayingDB3::MapEditor Class Reference . . . 40
7.11.1 Detailed Description . . . 40
7.11.2 Constructor & Destructor Documentation . . . 40
7.11.2.1 MapEditor() . . . 40
7.11.3 Member Function Documentation . . . 41
7.11.3.1 _createEditDialog() . . . 41
7.11.3.2 close() . . . 41
7.11.3.3 drawspace() . . . 41
7.11.3.4 edit() . . . 42
7.11.3.5 new() [1/2] . . . 42
7.11.3.6 new() [2/2] . . . 42
7.11.3.7 open() . . . 43
7.11.3.8 openfile() . . . 43
7.11.3.9 opennew() . . . 43
7.11.3.10 openold() . . . 43

7.11.3.11 print() . . . 44
7.11.3.12 recreateXML() . . . 44
7.11.3.13 save() . . . 44
7.11.3.14 saveas() . . . 44
7.11.3.15 scaleXY() . . . 44
7.11.3.16 unscaleXY() . . . 45
7.11.3.17 updateleveltree() . . . 45
7.11.3.18 updatemediatree() . . . 46
7.12 ParseXML Class Reference . . . 46
7.12.1 Detailed Description . . . 46
7.12.2 Constructor & Destructor Documentation . . . 46
7.12.2.1 ParseXML() . . . 46
7.12.2.2 ~ParseXML() . . . 47
7.12.3 Member Function Documentation . . . 47
7.12.3.1 displayTree() . . . 47
7.12.3.2 TopContainer() . . . 47
7.12.3.3 validate() . . . 48
7.13 RolePlayingDB3::PrintDialog Class Reference . . . 48
7.13.1 Detailed Description . . . 48
7.13.2 Member Function Documentation . . . 49
7.13.2.1 printprogress_end() . . . 49
7.13.2.2 printprogress_setpageno() . . . 49
7.13.2.3 printprogress_start() . . . 49
7.14 vfs::rpg::RPGDirent Class Reference . . . 49
7.14.1 Detailed Description . . . 50
7.14.2 Constructor & Destructor Documentation . . . 50
7.14.2.1 RPGDirent() . . . 50
7.14.2.2 ~RPGDirent() . . . 51

7.14.3 Member Function Documentation . . . 51
7.14.3.1 AddNewDirent() . . . 51
7.14.3.2 GetDirents() . . . 52
7.14.3.3 LookupMatch() . . . 52
7.14.3.4 LookupName() . . . 52
7.14.3.5 NumDirents() . . . 53
7.14.3.6 RemoveDirent() . . . 53
7.15 vfs::rpg::RPGFileSystem Class Reference . . . 54
7.15.1 Detailed Description . . . 54
7.15.2 Constructor & Destructor Documentation . . . 54
7.15.2.1 RPGFileSystem() . . . 54
7.15.2.2 ∼RPGFileSystem() . . . 55
7.16 RolePlayingDB3::ScrolledCanvas Class Reference . . . 55
7.16.1 Detailed Description . . . 55
7.16.2 Constructor & Destructor Documentation . . . 55
7.16.2.1 ScrolledCanvas() . . . 55
7.16.3 Member Function Documentation . . . 56
7.16.3.1 bindcanvas() . . . 56
7.17 RolePlayingDB3::ScrolledList Class Reference . . . 56
7.17.1 Detailed Description . . . 57
7.17.2 Constructor & Destructor Documentation . . . 57
7.17.2.1 ScrolledList() . . . 57
7.17.3 Member Function Documentation . . . 57
7.17.3.1 insert() . . . 57
7.17.3.2 itemcget() . . . 58
7.17.3.3 items() . . . 58
7.18 RolePlayingDB3::SheetEdit Class Reference . . . 58
7.18.1 Detailed Description . . . 59

7.18.2 Constructor & Destructor Documentation . . . 59
7.18.2.1 SheetEdit() . . . 59
7.18.3 Member Function Documentation . . . 59
7.18.3.1 close() . . . 59
7.18.3.2 edit() . . . 60
7.18.3.3 getfile() . . . 60
7.18.3.4 new() [1/2] . . . 60
7.18.3.5 new() [2/2] . . . 61
7.18.3.6 open() [1/2] . . . 61
7.18.3.7 open() [2/2] . . . 61
7.18.3.8 openfile() . . . 61
7.18.3.9 opennew() . . . 62
7.18.3.10 openold() . . . 62
7.18.3.11 print() . . . 62
7.18.3.12 recreateXML() . . . 62
7.18.3.13 save() . . . 63
7.18.3.14 saveas() . . . 63
7.19 SimpleDOMElement Class Reference . . . 63
7.19.1 Detailed Description . . . 64
7.19.2 Constructor & Destructor Documentation . . . 64
7.19.2.1 SimpleDOMElement() . . . 64
7.19.2.2 ~SimpleDOMElement() . . . 64
7.19.3 Member Function Documentation . . . 65
7.19.3.1 addchild() . . . 65
7.19.3.2 appenddata() . . . 65
7.19.3.3 attribute() . . . 65
7.19.3.4 children() . . . 66
7.19.3.5 data() . . . 66

7.19.3.6 display() . . . 66
7.19.3.7 getElementsById() . . . 67
7.19.3.8 getElementsByTagName() . . . 67
7.19.3.9 getParent() . . . 67
7.19.3.10 isChild() . . . 68
7.19.3.11 length() . . . 68
7.19.3.12 removeChild() . . . 68
7.19.3.13 replaceChildren() . . . 69
7.19.3.14 setAttribute() . . . 69
7.19.3.15 setdata() . . . 69
7.19.3.16 validate() . . . 70
7.20 RolePlayingDB3::SpaceEditor Class Reference . . . 70
7.20.1 Detailed Description . . . 71
7.20.2 Constructor & Destructor Documentation . . . 71
7.20.2.1 SpaceEditor() . . . 71
7.20.3 Member Function Documentation . . . 71
7.20.3.1 checksave() . . . 71
7.20.3.2 close() . . . 72
7.20.3.3 colorchanged() . . . 72
7.20.3.4 getcolor() . . . 72
7.20.3.5 getspacefile() . . . 72
7.20.3.6 new() [1/2] . . . 73
7.20.3.7 new() [2/2] . . . 73
7.20.3.8 open() [1/2] . . . 73
7.20.3.9 open() [2/2] . . . 73
7.20.3.10 print() . . . 74
7.20.3.11 printSpace() . . . 74
7.20.3.12 recreateXML() . . . 75

7.20.3.13 redrawspace() . . . 75
7.20.3.14 save() . . . 75
7.20.3.15 saveas() . . . 75
7.20.3.16 setdirty() . . . 76
7.21 RolePlayingDB3::Template Class Reference . . . 76
7.21.1 Detailed Description . . . 76
7.21.2 Constructor & Destructor Documentation . . . 76
7.21.2.1 Template() . . . 76
7.21.3 Member Function Documentation . . . 77
7.21.3.1 close() . . . 77
7.21.3.2 edit() . . . 77
7.21.3.3 getfile() . . . 77
7.21.3.4 new() [1/2] . . . 77
7.21.3.5 new() [2/2] . . . 78
7.21.3.6 newPDFPage() . . . 78
7.21.3.7 open() [1/2] . . . 78
7.21.3.8 open() [2/2] . . . 79
7.21.3.9 openfile() . . . 79
7.21.3.10 opennew() . . . 79
7.21.3.11 openold() . . . 79
7.21.3.12 outputXMLToPDF() . . . 80
7.21.3.13 print() . . . 80
7.21.3.14 rescantemplates() . . . 80
7.21.3.15 save() . . . 80
7.21.3.16 saveas() . . . 80
7.22 RolePlayingDB3::XMLContentEditor Class Reference . . . 81
7.22.1 Detailed Description . . . 81
7.22.2 Constructor & Destructor Documentation . . . 81
7.22.2.1 XMLContentEditor() . . . 81
7.22.2.2 ~XMLContentEditor() . . . 82
7.22.3 Member Function Documentation . . . 82
7.22.3.1 bcmdmethod() . . . 82
7.22.3.2 ExtractTagValue() . . . 83
7.22.3.3 getElementWidgetById() . . . 83
7.22.3.4 getlineno() . . . 83
7.22.3.5 getpageno() . . . 84
7.22.3.6 outputXMLToPDF() . . . 84
7.22.3.7 recreateXML() . . . 84

Bibliography 85

Index 85

Chapter 1

GNU GENERAL PUBLIC LICENSE

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software–to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

```
                GNU GENERAL PUBLIC LICENSE
```

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

1. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

```
a) You must cause the modified files to carry prominent notices
stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in
whole or in part contains or is derived from the Program or any
part thereof, to be licensed as a whole at no charge to all third
parties under the terms of this License.

c) If the modified program normally reads commands interactively
when run, you must cause it, when started running for such
interactive use in the most ordinary way, to print or display an
announcement including an appropriate copyright notice and a
notice that there is no warranty (or else, saying that you provide
a warranty) and that users may redistribute the program under
these conditions, and telling the user how to view a copy of this
License.  (Exception: if the Program itself is interactive but
does not normally print such an announcement, your work based on
the Program is not required to print an announcement.)
```

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

1. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

```
a) Accompany it with the complete corresponding machine-readable
source code, which must be distributed under the terms of Sections
1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three
years, to give any third party, for a charge no more than your
cost of physically performing source distribution, a complete
machine-readable copy of the corresponding source code, to be
distributed under the terms of Sections 1 and 2 above on a medium
customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer
to distribute corresponding source code.  (This alternative is
allowed only for noncommercial distribution and only if you
received the program in object code or executable form with such
an offer, in accord with Subsection b above.)
```

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

1. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

1. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

    (a) If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

1. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

1. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

1. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

1. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PR↩ OGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WIT↩ HOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSE↩ QUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year  name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items–whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

TclCommon 13
RPGSupport 14

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

gettext 15
ReadConfiguration 16
RolePlayingDB3 18
vfs::rpg 21

Chapter 4

Class Index

4.1 Class List

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

RolePlayingDB3::chroot_chooseDirectory . . . 23
RolePlayingDB3::chroot_getFile . . . 23
Dialog . . . 23
DynamicHelp . . . 24
RolePlayingDB3::Graphic . . . 26
IconBitmap . . . 27
IconImage . . . 28
RolePlayingDB3::LabeledDirTree . . . 28
RolePlayingDB3::LabeledScrolledText . . . 31
RolePlayingDB3::LevelEditor . . . 32
RolePlayingDB3::MapEditor . . . 40
ParseXML
Class to hold an XML tree. This class parses an XML string and stores the result as a DOM Element tree . . . 46
RolePlayingDB3::PrintDialog . . . 48
vfs::rpg::RPGDirent
A Directory Entry Holds information about a file or directory. Only the bare minimum information is kept: name, mod time, list of files (directories only), mode (permissions), offset (files only), and size (files only) . . . 49
vfs::rpg::RPGFileSystem
The filesystem class. This is the class that implements a simple Virtual Filesystem for internal use by the Roleplaying Database program. It stores the working version of the RPG DB bundle which is normally stored as a ZIP file. It is a basic and simple file system, using a backing file as temporary file system storage . . . 54
RolePlayingDB3::ScrolledCanvas . . . 55
RolePlayingDB3::ScrolledList . . . 56
RolePlayingDB3::SheetEdit . . . 58
SimpleDOMElement
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 . . . 63
RolePlayingDB3::SpaceEditor . . . 70
RolePlayingDB3::Template . . . 76
RolePlayingDB3::XMLContentEditor . . . 81

Chapter 5

Module Documentation

5.1 TclCommon

Namespaces

- gettext
- ReadConfiguration

Classes

- class DynamicHelp
- class IconImage
- class IconBitmap
- class SimpleDOMElement

  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.
- class ParseXML

  Class to hold an XML tree. This class parses an XML string and stores the result as a DOM Element tree.

Functions

- DynamicHelp_include (type)

5.1.1 Detailed Description

5.1.2 Function Documentation

5.1.2.1 DynamicHelp_include()

```
DynamicHelp_include (
            type  )
```

Macro to add Dynamic Help Add options and support for Dynamic Help to widgets.

5.2 RPGSupport

Namespaces

- vfs::rpg
- RolePlayingDB3

5.2.1 Detailed Description

Chapter 6

Namespace Documentation

6.1 gettext Namespace Reference

Functions

- _m (msgid,...)
- _mx (...)
- _ (...)

6.1.1 Detailed Description

Localization functions. Functions to fetch localized messages from the message catalog.

Author

Robert Heller <heller@deepsoft.com>

6.1.2 Package provided

gettext 1.0

6.1.3 Function Documentation

6.1.3.1 _()

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

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

6.1.3.2 _m()

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

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

6.1.3.3 _mx()

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

Get maxlength of a set of messages with a context hint prefix. Used to compute label widths.

6.2 ReadConfiguration Namespace Reference

Functions

- ReadConfiguration (filename, configurationArrayName)
- IsEven (i)
- WriteConfiguration (filename, configurationArrayName)
- ConfigurationType (...)

6.2.1 Detailed Description

The Read Configuration File code is contained in this namespace.

6.2.2 Function Documentation

6.2.2.1 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 defination 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 defination.

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.
- getanonoymous Gets the anonoymous option. Takes no arguments.

6.2.2.2 IsEven()

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

Check if a number is even.

Parameters

i	The number to check.

Returns

a boolean.

6.2.2.3 ReadConfiguration()

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

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

Parameters

filename	The name of the configuration file.
configurationArrayName	The name of the array to hold the configuration.

6.2.2.4 WriteConfiguration()

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

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

Parameters

filename	The name of the configuration file.
configurationArrayName	The name of the array holding the configuration.

6.3 RolePlayingDB3 Namespace Reference

Classes

- class chroot_chooseDirectory
- class chroot_getFile
- class Graphic
- class LabeledDirTree
- class LabeledScrolledText
- class LevelEditor
- class MapEditor
- class PrintDialog
- class ScrolledCanvas
- class ScrolledList
- class SheetEdit
- class SpaceEditor
- class Template
- class XMLContentEditor

Functions

- MapBundleMountPoint ()
- MapEditorOption ()
- LevelEditorOption ()
- LevelDirOption ()
- SpaceFileOption ()
- OpenFilename (optionspec="-openfilename openFilename OpenFilename")
- SaveFilename (optionspec="-savefilename saveFilename SaveFilename")
- GeneratePrintDialog (nameprefix, additionalcomps, createbody)

Variables

- **RolePlayingDB3**

6.3.1 Detailed Description

Contains all utility code specific to the RolePlayingDB3 system.

6.3.2 Function Documentation

6.3.2.1 GeneratePrintDialog()

```
RolePlayingDB3::GeneratePrintDialog (
            nameprefix ,
            additionalcomps ,
            createbody  )
```

Macro to generate code to create a print dialog.

Parameters

nameprefix	Dialog name prefix
additionalcomps	Additional components
createbody	Additional dialog create body.

6.3.2.2 LevelDirOption()

```
RolePlayingDB3::LevelDirOption ( )
```

Macro to add the -leveldir option

6.3.2.3 LevelEditorOption()

```
RolePlayingDB3::LevelEditorOption ( )
```

Macro to add the -leveleditor option

6.3.2.4 MapBundleMountPoint()

```
RolePlayingDB3::MapBundleMountPoint ( )
```

Macro to add the -mapbundlemountpoint option.

6.3.2.5 MapEditorOption()

```
RolePlayingDB3::MapEditorOption ( )
```

Macro to add the -mapeditor option.

6.3.2.6 OpenFilename()

```
RolePlayingDB3::OpenFilename (
            optionspec  = "-openfilename openFilename OpenFilename" )
```

Macro to add a readonly file option.

Parameters

optionspec	The option spec.

6.3.2.7 SaveFilename()

```
RolePlayingDB3::SaveFilename (
            optionspec  = "-savefilename saveFilename SaveFilename" )
```

Macro to add a read/write file option.

Parameters

optionspec	The option spec.

6.3.2.8 SpaceFileOption()

```
RolePlayingDB3::SpaceFileOption ( )
```

Macro to add -spacefile option.

6.4 vfs::rpg Namespace Reference

Classes

- class RPGDirent

  A Directory Entry Holds information about a file or directory. Only the bare minimum information is kept: name, mod time, list of files (directories only), mode (permissions), offset (files only), and size (files only).
- class RPGFileSystem

  The filesystem class. This is the class that implements a simple Virtual Filesystem for internal use by the Roleplaying Database program. It stores the working version of the RPG DB bundle which is normally stored as a ZIP file. It is a basic and simple file system, using a backing file as temporary file system storage.

Functions

- fail (code)
- Mount (mkfile, local,...)

6.4.1 Detailed Description

Roleplaying Database temp filesystem. This is an implementation of a simple Virtual Filesystem for internal use by the Roleplaying Database program. It stores the working version of the RPG DB bundle which is normally stored as a ZIP file. It is a basic and simple file system, using a backing file as temporary file system storage.

6.4.2 Function Documentation

6.4.2.1 fail()

```
vfs::rpg::fail (
            code  )
```

Generate a posixerror

Parameters

code	– the posix error to generate.

6.4.2.2 Mount()

```
vfs::rpg::Mount (
```

```
mkfile ,
local ,
 ...  )
```

Public function to mount a RPGFileSystem file system

Parameters

mkfile	Backing file
local	Mount point
...	Options: (none)

Returns

A RPGFileSystem object, bound to the mountpoint.

Chapter 7

Class Documentation

7.1 RolePlayingDB3::chroot_chooseDirectory Class Reference

7.1.1 Detailed Description

Chrooted version of tk_chooseDirectory. (copied and adapted from tk_chooseDirectory). This version is "chrooted" to the vfs holding a RPG bundle.

The documentation for this class was generated from the following file:

- RPGUtilities.tcl

7.2 RolePlayingDB3::chroot_getFile Class Reference

7.2.1 Detailed Description

Chrooted version of tk_getOpenFile / tk_getSaveFile. (copied and adapted from tk_getOpenFile / tk_getSaveFil). This version is "chrooted" to the vfs holding a RPG bundle.

The documentation for this class was generated from the following file:

- RPGUtilities.tcl

7.3 Dialog Class Reference

7.3.1 Detailed Description

delegate method add to bbox

The documentation for this class was generated from the following file:

- snitdialog.tcl

7.4 DynamicHelp Class Reference

Static Public Member Functions

- static configure (...)
- static register (path, _type,...)
- static add (path,...)
- static delete (path)

7.4.1 Detailed Description

Implements BWidget style Dynamic Help: balloon help and status variable help displays

7.4.2 Member Function Documentation

7.4.2.1 add()

```
static DynamicHelp::add (
             path ,
              ...  )  [static]
```

Add a path to the list of paths with Dynamic Help.

Parameters

path	The widget path.
...	Options: • -type The type of Dynamic Help. • -text The help text. • -item The menu item. • -index The item index. • -command The command. • -variable The variable.

7.4.2.2 configure()

```
static DynamicHelp::configure (
              ...  )  [static]
```

Set configuration options. ... Option value list: option value ...

- -state configure new state value.
- -delay configure new delay value.

7.4.2.3 delete()

```
static DynamicHelp::delete (
             path  )  [static]
```

Delete a path from Dynamic Help.

Parameters

path	The widget path.

7.4.2.4 register()

```
static DynamicHelp::register (
             path ,
             _type ,
              ...  )  [static]
```

Register a new widget with help.

Parameters

path	Widget path.
_type	Type of help (variable or balloon)
...	Additional args (depends on type).

The documentation for this class was generated from the following file:

- DynamicHelp.tcl

7.5 RolePlayingDB3::Graphic Class Reference

Public Member Functions

- getimage ()
- Graphic (...)

7.5.1 Detailed Description

Build a widget to fetch a graphcs file and display the image.

7.5.2 Constructor & Destructor Documentation

7.5.2.1 Graphic()

```
RolePlayingDB3::Graphic::Graphic (
               ...  )
```

Build a widget to fetch and display an image file.

Parameters

widgetpath	The widget path
...	Options: • -modifycmd Script to run when the widget is updated. • -text Filename in the file entry. • -basedirectory Base directory. • -label Label for the fileentry. • -labelwidth Label width for the fileentry. • -initialdir Initial dir for the fileentry.

7.5.3 Member Function Documentation

7.5.3.1 getimage()

```
RolePlayingDB3::Graphic::getimage ( )
```

Fetch the graphic image object.

Returns

an image object

The documentation for this class was generated from the following file:

- RPGUtilities.tcl

7.6 IconBitmap Class Reference

Static Public Member Functions

- static bitmap (name)

7.6.1 Detailed Description

Icon bitmap management Auto loads bitmaps and keeps track of available and loaded bitmap objects for labels and related uses.

7.6.2 Member Function Documentation

7.6.2.1 bitmap()

```
static IconBitmap::bitmap (
            name  )  [static]
```

Fetch a bitmap by name.

Parameters

name	Bitmap name.

The documentation for this class was generated from the following file:

- IconsAndImages.tcl

7.7 IconImage Class Reference

Static Public Member Functions

- static image (name,...)

7.7.1 Detailed Description

Icon image management Auto loads images and keeps track of available and loaded image objects for labels and related uses.

7.7.2 Member Function Documentation

7.7.2.1 image()

```
static IconImage::image (
             name ,
              ...  )  [static]
```

Fetch an image by name.

Parameters

name	Image name.
...	Options: • -filetype File type. • -icondir Alternitive directory to look in.

The documentation for this class was generated from the following file:

- IconsAndImages.tcl

7.8 RolePlayingDB3::LabeledDirTree Class Reference

Public Member Functions

- itemcget (node, option)
- redrawdirtree ()
- binditem (what, sequence, script)
- LabeledDirTree (...)

7.8.1 Detailed Description

Labeled Directory Tree Widget

7.8.2 Constructor & Destructor Documentation

7.8.2.1 LabeledDirTree()

```
RolePlayingDB3::LabeledDirTree::LabeledDirTree (
               ...  )
```

Constructor. Build the widget.

Parameters

widgetpath	The widget path.
...	Options: • -auto Passed to the scrolling window. • -scrollbar Passed to the scrolling window. • -showextension Show the file's extension. • -directory The root directory. • -filepattern The file pattern to use. • -sortfunction Sorting function. • -opendirs Open dirs initially or not. • -nofiles Controls whether to include files or not. • -label Label text for the widget.

7.8.3 Member Function Documentation

7.8.3.1 binditem()

```
RolePlayingDB3::LabeledDirTree::binditem (
            what ,
            sequence ,
            script  )
```

Bind events to items

Parameters

what	What tag to bind to.
sequence	Sequence to bind.
script	Strict to bind.

7.8.3.2 itemcget()

```
RolePlayingDB3::LabeledDirTree::itemcget (
            node ,
            option  )
```

Gets an option from an item.

Parameters

node	Item to get an option from.
option	The option to fetch.

Returns

The option value.

Exceptions

Unsupported-Option	For unsupported options.

7.8.3.3 redrawdirtree()

```
RolePlayingDB3::LabeledDirTree::redrawdirtree ( )
```

Redraw directory tree

The documentation for this class was generated from the following file:

- RPGUtilities.tcl

7.9 RolePlayingDB3::LabeledScrolledText Class Reference

Public Member Functions

- bind (...)
- LabeledScrolledText (...)

7.9.1 Detailed Description

Labeled Scrolled Text widget

7.9.2 Constructor & Destructor Documentation

7.9.2.1 LabeledScrolledText()

```
RolePlayingDB3::LabeledScrolledText::LabeledScrolledText (
              ...  )
```

Constructor

Parameters

widgetpath	
...	Options: • -label Label for frame • -auto Passed to Scrolled Window • -scrollbar Passed to Scrolled Window • -text Get/Set text in text widget

any options a text widget takes except -xscrollcommand and

-yscrollcommand All text methods (except xview and yview) are available as well.

7.9.3 Member Function Documentation

7.9.3.1 bind()

```
RolePlayingDB3::LabeledScrolledText::bind (
               ...  )
```

Bind events to the text widget.

Parameters

...	Arguments passed to bind.

The documentation for this class was generated from the following file:

- RPGUtilities.tcl

7.10 RolePlayingDB3::LevelEditor Class Reference

Public Member Functions

- checksave ()
- new ()
- open ()
- save ()
- saveas (_filename="")
- print ()
- close (...)
- LevelEditor (...)
- updatespacetree ()
- drawspace (canvas, X, Y, color, size,...)
- updatelevelmapcolor (color, spacetag)
- rescalelevelmap ()
- drawonespace (X, Y, color, spacetag)
- updatelevelmap ()
- recreateXML ()
- updatemediatree ()
- removeeditor (se)
- updateleveltree ()
- updatemediatree ()
- setspaceposition (XLE, YLE, spacename)

Static Public Member Functions

- static new (...)
- static open (...)
- static printLevel (pdfobj, leveldir, curpageV, curlineV, heading,...)

7.10.1 Detailed Description

Level editor

7.10.2 Constructor & Destructor Documentation

7.10.2.1 LevelEditor()

```
RolePlayingDB3::LevelEditor::LevelEditor (
              ...  )
```

Constructor – construct a level editor.

Parameters

widgetpath	Widget path.
...	Options: • -template Template to use for a new level. • -mapbundlemountpoint Map bundle mount point. • -mapeditor Parent map editor. • -leveldir Level directory.

7.10.3 Member Function Documentation

7.10.3.1 checksave()

```
RolePlayingDB3::LevelEditor::checksave ( )
```

Check if save is needed.

7.10.3.2 close()

```
RolePlayingDB3::LevelEditor::close (
              ...  )
```

Close this level editor.

Parameters

...	Options: • -dontask Flag to supress asking. • -dontsave Flag to supress saving. • -closingallwindows Flag if we are closing all windows.

7.10.3.3 drawonespace()

```
RolePlayingDB3::LevelEditor::drawonespace (
             X ,
             Y ,
             color ,
             spacetag  )
```

Draw one space

Parameters

X	The X coordinate.
Y	The Y coordinate.
color	The color.
spacetag	The space tag.

7.10.3.4 drawspace()

```
RolePlayingDB3::LevelEditor::drawspace (
             canvas ,
             X ,
             Y ,
             color ,
             size ,
              ...  )
```

Draw the space.

Parameters

canvas	The canvas to draw to.
X	The X coordinate.
Y	The Y coordinate.
color	The color.
size	The size.
...	Options passed along to the map editor draw space method.

7.10.3.5 new() [1/2]

```
RolePlayingDB3::LevelEditor::new ( )
```

Create a new level.

7.10.3.6 new() [2/2]

```
static RolePlayingDB3::LevelEditor::new (
              ...  )  [static]
```

Create a new level.

Parameters

...	Options: • -mapbundlemountpoint Map mount point bundle. • -parent Parent window. • -mapeditor Parent map editor. • -leveldir Level directory.

Returns

the new toplevel.

7.10.3.7 open() [1/2]

```
RolePlayingDB3::LevelEditor::open ( )
```

Open a level file.

7.10.3.8 open() [2/2]

```
static RolePlayingDB3::LevelEditor::open (
              ... ) [static]
```

Open a level file.

Parameters

...	Options: • -mapbundlemountpoint Map mount point bundle. • -parent Parent window. • -mapeditor Parent map editor. • -leveldir Level directory.

Returns

the new toplevel.

7.10.3.9 print()

```
RolePlayingDB3::LevelEditor::print ( )
```

Print this level (to a PDF file).

7.10.3.10 printLevel()

```
static RolePlayingDB3::LevelEditor::printLevel (
             pdfobj ,
             leveldir ,
             curpageV ,
             curlineV ,
             heading ,
              ... ) [static]
```

Print (to a PDF file) the level.

Parameters

pdfobj	The PDF object to print to.
leveldir	The level directory.
curpageV	Current page variable.

Parameters

curlineV	Current line variable.
heading	Heading.
...	Options: • -mapbundlemountpoint The map bundle mount point. • -mapeditor The parent map editor. • -parent The parent window.

7.10.3.11 recreateXML()

```
RolePlayingDB3::LevelEditor::recreateXML ( )
```

Regenerate the XML string.

7.10.3.12 removeeditor()

```
RolePlayingDB3::LevelEditor::removeeditor (
            se  )
```

Remove the selected editor.

Parameters

se	The space editor to remove.

7.10.3.13 rescalelevelmap()

```
RolePlayingDB3::LevelEditor::rescalelevelmap ( )
```

Rescale the level map.

7.10.3.14 save()

```
RolePlayingDB3::LevelEditor::save ( )
```

Save the current file.

7.10.3.15 saveas()

```
RolePlayingDB3::LevelEditor::saveas (
            _filename  = "" )
```

Save to a new file.

Parameters

_filename	The filename to save to.

7.10.3.16 setspaceposition()

```
RolePlayingDB3::LevelEditor::setspaceposition (
            XLE ,
            YLE ,
            spacename  )
```

Set the space's position.

Parameters

XLE	The X position entry.
YLE	The Y position entry.
spacename	The space name.

7.10.3.17 updatelevelmap()

```
RolePlayingDB3::LevelEditor::updatelevelmap ( )
```

Update the level map.

7.10.3.18 updatelevelmapcolor()

```
RolePlayingDB3::LevelEditor::updatelevelmapcolor (
            color ,
            spacetag  )
```

Update the level map color.

Parameters

color	New color
spacetag	The space tag.

7.10.3.19 updateleveltree()

```
RolePlayingDB3::LevelEditor::updateleveltree ( )
```

Update the level tree.

7.10.3.20 updatemediatree() [1/2]

```
RolePlayingDB3::LevelEditor::updatemediatree ( )
```

Update the media tree.

7.10.3.21 updatemediatree() [2/2]

```
RolePlayingDB3::LevelEditor::updatemediatree ( )
```

Update the media tree.

7.10.3.22 updatespacetree()

```
RolePlayingDB3::LevelEditor::updatespacetree ( )
```

Update space tree.

The documentation for this class was generated from the following file:

- RPGMapLevelSpace.tcl

7.11 RolePlayingDB3::MapEditor Class Reference

Public Member Functions

- scaleXY (X, Y, scale)
- unscaleXY (cX, cY, scale)
- drawspace (canvas, X, Y, color, size,...)
- opennew ()
- new ()
- openold (_filename)
- save ()
- saveas (_filename="")
- print ()
- close (...)
- MapEditor (...)
- recreateXML ()
- updateleveltree ()
- updatemediatree ()

Static Public Member Functions

- static _createEditDialog ()
- static edit (...)
- static new (...)
- static open (...)
- static openfile (filename)

7.11.1 Detailed Description

Map editor widget.

7.11.2 Constructor & Destructor Documentation

7.11.2.1 MapEditor()

```
RolePlayingDB3::MapEditor::MapEditor (
              ...  )
```

Constructor – create a map edit widget.

Parameters

widgetpath	Widget path.
...	Options: • -template Alternitive emplate to use. • -openfilename File to open.

7.11.3 Member Function Documentation

7.11.3.1 _createEditDialog()

```
static RolePlayingDB3::MapEditor::_createEditDialog ( )  [static]
```

Create edit dialog.

7.11.3.2 close()

```
RolePlayingDB3::MapEditor::close (
              ...  )
```

Close the current file.

Parameters

...	Options: None.

7.11.3.3 drawspace()

```
RolePlayingDB3::MapEditor::drawspace (
            canvas ,
            X ,
            Y ,
            color ,
            size ,
             ...  )
```

Draw a space.

Parameters

canvas	The canvas to draw on.
X	X coord.
Y	Y coord.
color	The color of the space.
size	The size of the space.
...	Additional args passed to canvas create.

7.11.3.4 edit()

```
static RolePlayingDB3::MapEditor::edit (
              ... ) [static]
```

Edit an existing map.

Parameters

...	Options: None.

7.11.3.5 new() [1/2]

```
static RolePlayingDB3::MapEditor::new (
              ... ) [static]
```

Create a new map.

Parameters

...	Options: • -template Template to use. • -parent The parent window.

7.11.3.6 new() [2/2]

```
RolePlayingDB3::MapEditor::new ( )
```

Create a new map.

7.11.3.7 open()

```
static RolePlayingDB3::MapEditor::open (
              ...  )  [static]
```

Open an existing map bundle file.

Parameters

...	Options: • -parent Parent window. • -like Open a file like this one.

7.11.3.8 openfile()

```
static RolePlayingDB3::MapEditor::openfile (
             filename  )  [static]
```

Open an existing map file.

Parameters

filename	File to open.

7.11.3.9 opennew()

```
RolePlayingDB3::MapEditor::opennew ( )
```

Create a new map.

7.11.3.10 openold()

```
RolePlayingDB3::MapEditor::openold (
             _filename  )
```

Open an old file.

Parameters

_filename	Filename to open.

7.11.3.11 print()

```
RolePlayingDB3::MapEditor::print ( )
```

Print (to a PDF file) the current map.

7.11.3.12 recreateXML()

```
RolePlayingDB3::MapEditor::recreateXML ( )
```

Regenerate the XML string.

7.11.3.13 save()

```
RolePlayingDB3::MapEditor::save ( )
```

Save the current file.

7.11.3.14 saveas()

```
RolePlayingDB3::MapEditor::saveas (
            _filename  = "" )
```

Save the bundle to the speified file.

Parameters

_filename	Filename to save to.

7.11.3.15 scaleXY()

```
RolePlayingDB3::MapEditor::scaleXY (
            X ,
```

```
            Y ,
            scale  )
```

Scale space on map.

Parameters

X	X coord.
Y	Y coord.
scale	Scale factor.

Returns

Coordlist, scaled and tranlated.

7.11.3.16 unscaleXY()

```
RolePlayingDB3::MapEditor::unscaleXY (
            cX ,
            cY ,
            scale  )
```

Unscale space on map.

Parameters

cX	X canvas coord.
cY	Y canvas coord.
scale	Scale factor.

Returns

Coord list, unscaled and translated.

7.11.3.17 updateleveltree()

```
RolePlayingDB3::MapEditor::updateleveltree ( )
```

Update the level tree.

7.11.3.18 updatemediatree()

```
RolePlayingDB3::MapEditor::updatemediatree ( )
```

Update the media tree.

The documentation for this class was generated from the following file:

- RPGMapLevelSpace.tcl

7.12 ParseXML Class Reference

Class to hold an XML tree. This class parses an XML string and stores the result as a DOM Element tree.

Public Member Functions

- ParseXML (xml,...)
- ~ParseXML ()
- displayTree (fp="stdout",...)

Static Public Member Functions

- static validate (object)

  Validation typemethod. Raises an error if its argument is not a ParseXML object.
- static TopContainer (XML)

7.12.1 Detailed Description

Class to hold an XML tree. This class parses an XML string and stores the result as a DOM Element tree.

Inheirets methods getElementsByTagName, getElementsById, and children from SimpleDOMElement.

7.12.2 Constructor & Destructor Documentation

7.12.2.1 ParseXML()

```
ParseXML::ParseXML (
             xml ,
              ...  )
```

The constructor parses the XML string and stores it as a child of the rootnode component.

Parameters

objname	Generally %AUTO% is passed.
xml	The XML string.
...	Options. None at present.

7.12.2.2 ~ParseXML()

```
ParseXML::~ParseXML ( )
```

The destructor – free up memory by destroying the root node.

7.12.3 Member Function Documentation

7.12.3.1 displayTree()

```
ParseXML::displayTree (
             fp  = "stdout",
              ...  )
```

Display the XML tree.

Parameters

fp	The channel to write the display to.
...	Options • -addnamespace Boolean (default no) Add namespace.

7.12.3.2 TopContainer()

```
static ParseXML::TopContainer (
             XML  )  [static]
```

Static function to peek into XML to find the top container.

Parameters

XML	XML string.

Returns

The top container tag.

7.12.3.3 validate()

```
static ParseXML::validate (
            object  )  [static]
```

Validation typemethod. Raises an error if its argument is not a ParseXML object.

Parameters

object	The object to typecheck.

Returns

The object or raise an error.

The documentation for this class was generated from the following file:

- ParseXML.tcl

7.13 RolePlayingDB3::PrintDialog Class Reference

Static Public Member Functions

- static printprogress_start (...)
- static printprogress_setpageno (pageno)
- static printprogress_end ()

7.13.1 Detailed Description

General purpose Print Dialog This is a general purpose print dialog.

7.13.2 Member Function Documentation

7.13.2.1 printprogress_end()

```
static RolePlayingDB3::PrintDialog::printprogress_end ( )  [static]
```

End the progress dialog: release grab and focus and enable the dismis button.

7.13.2.2 printprogress_setpageno()

```
static RolePlayingDB3::PrintDialog::printprogress_setpageno (
            pageno  )  [static]
```

Set the current print progress page number.

Parameters

pageno	New current page number.

7.13.2.3 printprogress_start()

```
static RolePlayingDB3::PrintDialog::printprogress_start (
              ...  )  [static]
```

Start the progress dialog.

Parameters

...	Options: • -parent Parent window.

The documentation for this class was generated from the following file:

- RPGUtilities.tcl

7.14 vfs::rpg::RPGDirent Class Reference

A Directory Entry Holds information about a file or directory. Only the bare minimum information is kept: name, mod time, list of files (directories only), mode (permissions), offset (files only), and size (files only).

Public Member Functions

- RPGDirent (name,...)
- ~RPGDirent ()
- AddNewDirent (name,...)
- RemoveDirent (dirent)
- GetDirents ()
- NumDirents ()
- LookupName (name)
- LookupMatch (pattern, types="file directory")

7.14.1 Detailed Description

A Directory Entry Holds information about a file or directory. Only the bare minimum information is kept: name, mod time, list of files (directories only), mode (permissions), offset (files only), and size (files only).

Options:

- -ftype This readonly option flags type of directory entry, file or directory.

Instance variables:

- _name The file name. Accessors: Name and SetName.
- _mtime The mod time. Accessors: MTime and SetMTime.
- _files The list of files (directories only). No direct accessors.
- _mode The file mode (permissions). Accessors Mode and SetMode.
- _offset The offset in the backing file (files only). Accessors Offset and SetOffset.
- _size The size of the file (files only). Accessors Size and SetSize

7.14.2 Constructor & Destructor Documentation

7.14.2.1 RPGDirent()

```
vfs::rpg::RPGDirent::RPGDirent (
            name ,
             ...  )
```

Constructor: create a new direct. Not normally called directly (see AddNewDirent).

Parameters

objname	Should be passed as AUTO%
name	New filename
...	Options: • -ftype File type. Must be either "file" or "directory". Readonly and defaults to "file".

7.14.2.2 ~RPGDirent()

```
vfs::rpg::RPGDirent::~RPGDirent ( )
```

Destructor – destroy the directory entry and all of its child directory entries.

7.14.3 Member Function Documentation

7.14.3.1 AddNewDirent()

```
vfs::rpg::RPGDirent::AddNewDirent (
            name ,
             ...  )
```

Add a new directory entry to this directory.

Parameters

name	The new filename.
...	Options: • -ftype The type of new directory entry to create. Must be one of "file" or "directory" and is "file" by default.

Returns

The new directory entry.

Exceptions

ENOENT	If the containing dirent is not a directory.

7.14.3.2 GetDirents()

```
vfs::rpg::RPGDirent::GetDirents ( )
```

Get a list of child dirents.

Returns

A list of dirents.

Exceptions

ENOENT	If the containing dirent is not a directory.

7.14.3.3 LookupMatch()

```
vfs::rpg::RPGDirent::LookupMatch (
            pattern ,
            types  = "file directory" )
```

LookupMatch using a glob pattern

Parameters

pattern	The (glob) pattern to look for.
types	The file types to check for. Must be a list containing "file" and/or "directory"

Returns

A list of dirents.

Exceptions

ENOENT	If the containing dirent is not a directory.

7.14.3.4 LookupName()

```
vfs::rpg::RPGDirent::LookupName (
            name  )
```

Lookup a filename (exact match).

Parameters

name	The filename to look for.

Returns

A dirent or {}.

Exceptions

ENOENT	If the containing dirent is not a directory.

7.14.3.5 NumDirents()

```
vfs::rpg::RPGDirent::NumDirents ( )
```

Get a count of child dirents.

Returns

The number of child dirents.

Exceptions

ENOENT	If the containing dirent is not a directory.

7.14.3.6 RemoveDirent()

```
vfs::rpg::RPGDirent::RemoveDirent (
            dirent  )
```

Remove a directory entry from this directory entry. The dirent is not destroyed! The calling function needs to take care of that or suffer a memory leak.

Parameters

dirent	The dirent to remove.

Returns

The dirent.

Exceptions

ENOENT	If the containing dirent is not a directory. (No error is reported if the dirent being removed is not in this dirent.)

The documentation for this class was generated from the following file:

- RPGFS.tcl

7.15 vfs::rpg::RPGFileSystem Class Reference

The filesystem class. This is the class that implements a simple Virtual Filesystem for internal use by the Roleplaying Database program. It stores the working version of the RPG DB bundle which is normally stored as a ZIP file. It is a basic and simple file system, using a backing file as temporary file system storage.

Public Member Functions

- RPGFileSystem (backingfile,...)
- ~RPGFileSystem ()

7.15.1 Detailed Description

The filesystem class. This is the class that implements a simple Virtual Filesystem for internal use by the Roleplaying Database program. It stores the working version of the RPG DB bundle which is normally stored as a ZIP file. It is a basic and simple file system, using a backing file as temporary file system storage.

7.15.2 Constructor & Destructor Documentation

7.15.2.1 RPGFileSystem()

```
vfs::rpg::RPGFileSystem::RPGFileSystem (
            backingfile ,
             ...  )
```

Constructor – takes care of mounting the file system. The root directory is set up. If the backing file already exists, the root directory is read in from the backing file.

Parameters

mountpoint	The name of the mount point.
backingfile	The name of the backing file. It is created if it does not exist.

7.15.2.2 ~RPGFileSystem()

```
vfs::rpg::RPGFileSystem::~RPGFileSystem ( )
```

Destructor – unmount the filesystem and free up all allocated memory. Flush the directory tree to the backing file if it is "dirty".

The documentation for this class was generated from the following file:

- RPGFS.tcl

7.16 RolePlayingDB3::ScrolledCanvas Class Reference

Public Member Functions

- ScrolledCanvas (...)
- bindcanvas (...)

7.16.1 Detailed Description

Scrolled Canvas Just a plain canvas, with scrollbars. All of the canvas's methods are exposed except xview yview.

7.16.2 Constructor & Destructor Documentation

7.16.2.1 ScrolledCanvas()

```
RolePlayingDB3::ScrolledCanvas::ScrolledCanvas (
              ...  )
```

Constructor – build a scrolled canvas.

Parameters

widgetpath	The widget path
...	Options: • -auto Passed to scroll window. • -scrollbar Passed to scroll window.

All of the options (except -xscrollcommand -yscrollcommand)

that the canvas command takes.

7.16.3 Member Function Documentation

7.16.3.1 bindcanvas()

```
RolePlayingDB3::ScrolledCanvas::bindcanvas (
              ...  )
```

Feed through bind to the canvas

Parameters

...	See 3tk bind

The documentation for this class was generated from the following file:

- RPGUtilities.tcl

7.17 RolePlayingDB3::ScrolledList Class Reference

Public Member Functions

- items ()
- itemcget (item, option)
- insert (where,...)
- ScrolledList (...)

7.17.1 Detailed Description

Scrolled Listbox many ttk::treeview methods passed through to the list, otherwise methods to simulate BWindget List are provided.

7.17.2 Constructor & Destructor Documentation

7.17.2.1 ScrolledList()

```
RolePlayingDB3::ScrolledList::ScrolledList (
               ...  )
```

Constructor – build a scrolled list box.

Parameters

widgetpath	
...	Options: • -auto Passed to scroll window. • -scrollbar Passed to scroll window.

All of the options (except -xscrollcommand, -yscrollcommand,

-show, -columns, and -displaycolumns) that the ttk::treeview command takes.

7.17.3 Member Function Documentation

7.17.3.1 insert()

```
RolePlayingDB3::ScrolledList::insert (
             where ,
              ...  )
```

Insert an item at a specified location.

Parameters

where	Index to insert at (end == at the end).
...	Options: passed through to ttk::treview's insert method.

7.17.3.2 itemcget()

```
RolePlayingDB3::ScrolledList::itemcget (
            item ,
            option  )
```

Get an item's property

Parameters

item	Item id.
option	The option to fetch.

Returns

the item property.

7.17.3.3 items()

```
RolePlayingDB3::ScrolledList::items ( )
```

Return a list of item ids in the list

Returns

a list of item ids.

The documentation for this class was generated from the following file:

- RPGUtilities.tcl

7.18 RolePlayingDB3::SheetEdit Class Reference

Public Member Functions

- getfile ()
- opennew ()
- new ()
- open ()
- openold (_filename)
- save ()
- saveas (_filename="")
- print ()
- close (...)
- SheetEdit (...)
- recreateXML ()

Static Public Member Functions

- static edit (...)
- static new (...)
- static open (...)
- static openfile (filename)

7.18.1 Detailed Description

Widget to edit a sheet. Wraps around XMLContentEditor to edit a "sheet". Gets installed into a standard RPG Toplevel frame – not normally installed directly.

7.18.2 Constructor & Destructor Documentation

7.18.2.1 SheetEdit()

```
RolePlayingDB3::SheetEdit::SheetEdit (
               ...  )
```

Constructor – create a sheet edit frame.

Parameters

widgetpath	The widget path.
...	Options: • -template Template to use (if this is a new sheet) • -sheetclass Sheet class (optional) • -openfilename Existing file to open (optional)

7.18.3 Member Function Documentation

7.18.3.1 close()

```
RolePlayingDB3::SheetEdit::close (
               ...  )
```

Close the current sheet.

Parameters

...	Options: none

7.18.3.2 edit()

```
static RolePlayingDB3::SheetEdit::edit (
              ...  )  [static]
```

Edit an existing sheet file.

Parameters

...	Options: • -sheetclass The sheet class to edit.

7.18.3.3 getfile()

```
RolePlayingDB3::SheetEdit::getfile ( )
```

the current file name

Returns

the file name.

7.18.3.4 new() [1/2]

```
static RolePlayingDB3::SheetEdit::new (
              ...  )  [static]
```

Create a new sheet and edit it.

Parameters

...	Options: • -template Sheet template to use. • -sheetclass Sheet class to create. • -parent Parent window.

7.18.3.5 new() [2/2]

```
RolePlayingDB3::SheetEdit::new ( )
```

Create a new sheet.

7.18.3.6 open() [1/2]

```
RolePlayingDB3::SheetEdit::open ( )
```

Open an existing sheet file.

7.18.3.7 open() [2/2]

```
static RolePlayingDB3::SheetEdit::open (
              ...  )  [static]
```

Open an existing sheet file.

Parameters

...	Options: • -sheetclass Sheet class to open. • -like Filename suggestion. • -parent Parent window.

7.18.3.8 openfile()

```
static RolePlayingDB3::SheetEdit::openfile (
```

```
            filename  )  [static]
```

Open an existing sheet file.

Parameters

filename	File to open.

7.18.3.9 opennew()

```
RolePlayingDB3::SheetEdit::opennew ( )
```

Open a new sheet file.

7.18.3.10 openold()

```
RolePlayingDB3::SheetEdit::openold (
            _filename  )
```

Open an old file.

Parameters

_filename	File to open.

7.18.3.11 print()

```
RolePlayingDB3::SheetEdit::print ( )
```

Print (write to PDF) the current sheet.

7.18.3.12 recreateXML()

```
RolePlayingDB3::SheetEdit::recreateXML ( )
```

Regenerate XML string file.

7.18.3.13 save()

```
RolePlayingDB3::SheetEdit::save ( )
```

Save the sheet to the current file.

7.18.3.14 saveas()

```
RolePlayingDB3::SheetEdit::saveas (
            _filename  = "" )
```

Save the sheet to a new file.

Parameters

_filename	File to save to.

The documentation for this class was generated from the following file:

- RPGSheetEdit.tcl

7.19 SimpleDOMElement Class Reference

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.

Public Member Functions

- SimpleDOMElement (...)
- ~SimpleDOMElement ()
- children ()
- replaceChildren (newchildren)
- addchild (childnode)
- length ()
- data ()
- setdata (d)
- appenddata (d)
- display (fp="stdout", indent="",...)
- attribute (attrname)
- setAttribute (attrname, value="")
- getElementsByTagName (thetag,...)
- getElementsById (theid)
- isChild (item)
- getParent (item)
- removeChild (item)

Static Public Member Functions

- static validate (object)

  Validation typemethod. Raises an error if its argument is not a SimpleDOMElement object.

7.19.1 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.

Author

Robert Heller <heller@deepsoft.com>.

7.19.2 Constructor & Destructor Documentation

7.19.2.1 SimpleDOMElement()

```
SimpleDOMElement::SimpleDOMElement (
               ...  )
```

The constructor. Just sets the options.

Parameters

objname	Element name. Generally %AUTO% is passed.
...	Options: • -tag The element's tag. • -attributes The element's attributes. • -opts The element's options.

7.19.2.2 ~SimpleDOMElement()

```
SimpleDOMElement::~SimpleDOMElement ( )
```

The destructor – free up all memory by destroying all children.

7.19.3 Member Function Documentation

7.19.3.1 addchild()

```
SimpleDOMElement::addchild (
            childnode  )
```

Method to add a child node.

Parameters

childnode	The child node to add.

7.19.3.2 appenddata()

```
SimpleDOMElement::appenddata (
            d  )
```

Method to append to the element's data.

Parameters

d	The new data.

7.19.3.3 attribute()

```
SimpleDOMElement::attribute (
            attrname  )
```

Method to return a selected attribute's value.

Parameters

attrname	The name of the attribute.

Returns

The attribute's value or the empty string.

7.19.3.4 children()

```
SimpleDOMElement::children ( )
```

Method to return the element's children.

Returns

The children.

7.19.3.5 data()

```
SimpleDOMElement::data ( )
```

Method to return the element's data.

Returns

The data.

7.19.3.6 display()

```
SimpleDOMElement::display (
            fp  = "stdout",
            indent  = "",
             ...  )
```

Method to display a node, along with its children, and a proper XML document.

Parameters

fp	Channel to write the display to.
indent	The indentation to use.
...	Options • -addnamespace Boolean (default no) Add namespace.

7.19.3.7 getElementsById()

```
SimpleDOMElement::getElementsById (
            theid  )
```

Method to return all of the elements under this element with the specified value of their id attribute.

Parameters

theid	The id value match.

Returns

A list of element object with the matching id value.

7.19.3.8 getElementsByTagName()

```
SimpleDOMElement::getElementsByTagName (
            thetag ,
             ...  )
```

Method to return all of the elements under this element with the specified tag name.

Parameters

thetag	The tag to match.

Returns

A list of element object with the matching tag.

7.19.3.9 getParent()

```
SimpleDOMElement::getParent (
            item  )
```

Method to get the parent of the item.

Parameters

item	The item to get the parent of.

Returns

The parent node or {} if none found.

7.19.3.10 isChild()

```
SimpleDOMElement::isChild (
            item  )
```

Method to check if the item is a child of this node.

Parameters

item	The possible child.

Returns

True if item is a child, false otherwise.

7.19.3.11 length()

```
SimpleDOMElement::length ( )
```

Method to return the number of children.

Returns

The number of children.

7.19.3.12 removeChild()

```
SimpleDOMElement::removeChild (
            item  )
```

Method to remove item from the children of this node.

Parameters

item	The item to remove.

7.19.3.13 replaceChildren()

```
SimpleDOMElement::replaceChildren (
            newchildren  )
```

Method to replace the element's children.

Parameters

newchildren	The new children list.

Returns

The old children

Exceptions

Invalid	type error if elements of newchildren are not SimpleDOMElement objects.

7.19.3.14 setAttribute()

```
SimpleDOMElement::setAttribute (
            attrname ,
            value  = "" )
```

Method to set a selected attribute's value.

Parameters

attrname	The name of the attribute.
value	The value to set. Default is the empty string.

7.19.3.15 setdata()

```
SimpleDOMElement::setdata (
            d  )
```

Method to set the element's data.

Parameters

d	The new data.

7.19.3.16 validate()

```
static SimpleDOMElement::validate (
            object  )  [static]
```

Validation typemethod. Raises an error if its argument is not a SimpleDOMElement object.

Parameters

object	The object to typecheck.

Returns

The object or raise an error.

The documentation for this class was generated from the following file:

- ParseXML.tcl

7.20 RolePlayingDB3::SpaceEditor Class Reference

Public Member Functions

- getspacefile ()
- setdirty ()
- checksave ()
- new ()
- open ()
- save ()
- saveas (_filename="")
- print ()
- close (...)
- SpaceEditor (...)
- colorchanged ()
- getcolor ()
- redrawspace ()
- recreateXML ()

Static Public Member Functions

- static new (...)
- static open (...)
- static printSpace (pdfobj, spacefile, curpageV, curlineV, heading,...)

7.20.1 Detailed Description

Space edit widget.

7.20.2 Constructor & Destructor Documentation

7.20.2.1 SpaceEditor()

```
RolePlayingDB3::SpaceEditor::SpaceEditor (
              ...  )
```

Constructor – construct a space edit widget.

Parameters

widgetpage	The widget path.
...	Options: • -template The template XML to use for a new space. • -mapbundlemountpoin The map bundle mount point. • -leveleditor The level editor. • -spacefile The space filename.

7.20.3 Member Function Documentation

7.20.3.1 checksave()

```
RolePlayingDB3::SpaceEditor::checksave ( )
```

Check if saving is needed.

7.20.3.2 close()

```
RolePlayingDB3::SpaceEditor::close (
              ... )
```

Close the current space.

Parameters

...	Options: • -dontask Flag to supress asking. • -dontsave Flag to supress saving. • -closingallwindows Flag to indicating that all windows are being closed.

7.20.3.3 colorchanged()

```
RolePlayingDB3::SpaceEditor::colorchanged ( )
```

Update on color change.

7.20.3.4 getcolor()

```
RolePlayingDB3::SpaceEditor::getcolor ( )
```

Get the current color.

Returns

the current color.

7.20.3.5 getspacefile()

```
RolePlayingDB3::SpaceEditor::getspacefile ( )
```

Retrieve the current space file.

Returns

the current space file.

7.20.3.6 new() [1/2]

```
RolePlayingDB3::SpaceEditor::new ( )
```

Create a new space.

7.20.3.7 new() [2/2]

```
static RolePlayingDB3::SpaceEditor::new (
              ... ) [static]
```

Create a new space.

Parameters

...	Options: • -mapbundlemountpoint The map bundle mount point. • -parent The parent window. • -leveleditor The level editor. • -leveldir The level directory. • -newspace The new space file.

Returns

the new toplevel.

7.20.3.8 open() [1/2]

```
RolePlayingDB3::SpaceEditor::open ( )
```

Open an existing space file.

7.20.3.9 open() [2/2]

```
static RolePlayingDB3::SpaceEditor::open (
              ... ) [static]
```

Open an existing space file.

Parameters

...	Options: • -mapbundlemountpoint The map bundle mount point. • -parent The parent window. • -leveleditor The level editor. • -leveldir The level directory. • -spacefile The new space file.

Returns

the new toplevel.

7.20.3.10 print()

```
RolePlayingDB3::SpaceEditor::print ( )
```

Print the current space.

7.20.3.11 printSpace()

```
static RolePlayingDB3::SpaceEditor::printSpace (
            pdfobj ,
            spacefile ,
            curpageV ,
            curlineV ,
            heading ,
             ... ) [static]
```

Print a space (to a PDF file).

Parameters

pdfobj	The PDF Object to print to.
spacefile	The space file to print.
curpageV	The current page variable.
curlineV	The current line variable.
heading	The heading.

Parameters

...	Options: • -mapbundlemountpoint The map bundle mount point. • -leveldir The level directory. • -parent The parent window. • -mapeditor The parent map editor.

7.20.3.12 recreateXML()

```
RolePlayingDB3::SpaceEditor::recreateXML ( )
```

Regenerate the XML string.

7.20.3.13 redrawspace()

```
RolePlayingDB3::SpaceEditor::redrawspace ( )
```

Redraw the space.

7.20.3.14 save()

```
RolePlayingDB3::SpaceEditor::save ( )
```

Save the current space to the current file.

7.20.3.15 saveas()

```
RolePlayingDB3::SpaceEditor::saveas (
            _filename  = "" )
```

Save the current space to a new file.

Parameters

_filename	The file name to save to.

7.20.3.16 setdirty()

```
RolePlayingDB3::SpaceEditor::setdirty ( )
```

Set the dirty flag.

The documentation for this class was generated from the following file:

- RPGMapLevelSpace.tcl

7.21 RolePlayingDB3::Template Class Reference

Public Member Functions

- getfile ()
- opennew ()
- new ()
- openold (_filename)
- rescantemplates ()
- open (...)
- save ()
- saveas (_filename="")
- print ()
- outputXMLToPDF (pdfobj, heading="Sheet")
- newPDFPage (pdfobj, heading, subheading)
- close (...)
- Template (...)

Static Public Member Functions

- static edit ()
- static new (...)
- static open (...)
- static openfile (filename)

7.21.1 Detailed Description

Template editor widget.

7.21.2 Constructor & Destructor Documentation

7.21.2.1 Template()

```
RolePlayingDB3::Template::Template (
              ...  )
```

Construct a template bundle editor widget.

Parameters

...	Options: • -template Not used.

7.21.3 Member Function Documentation

7.21.3.1 close()

```
RolePlayingDB3::Template::close (
              ... )
```

Close the current bundle file.

Parameters

...	Options: none.

7.21.3.2 edit()

```
static RolePlayingDB3::Template::edit ( ) [static]
```

Edit an existing template bundle.

7.21.3.3 getfile()

```
RolePlayingDB3::Template::getfile ( )
```

Get the current filename.

Returns

the current filename.

7.21.3.4 new() [1/2]

```
static RolePlayingDB3::Template::new (
              ... ) [static]
```

Create a new template bundle.

Parameters

...	Options: • -like Window to get a default filename from.

7.21.3.5 new() [2/2]

```
RolePlayingDB3::Template::new ( )
```

Create a new template bundle.

7.21.3.6 newPDFPage()

```
RolePlayingDB3::Template::newPDFPage (
            pdfobj ,
            heading ,
            subheading  )
```

Start a new page.

Parameters

pdfobj	The PDF object to print to.
heading	The heading.
subheading	The subheading.

7.21.3.7 open() [1/2]

```
static RolePlayingDB3::Template::open (
             ...  )  [static]
```

Open an existing template bundle.

Parameters

...	Options: • -like Editor to create a template bundle like.

7.21.3.8 open() [2/2]

```
RolePlayingDB3::Template::open (
              ... )
```

Open an existing template bundle.

7.21.3.9 openfile()

```
static RolePlayingDB3::Template::openfile (
             filename ) [static]
```

Open an existing template bundle file.

Parameters

filename	Name of the file to open.

7.21.3.10 opennew()

```
RolePlayingDB3::Template::opennew ( )
```

Open a new template bundle.

7.21.3.11 openold()

```
RolePlayingDB3::Template::openold (
             _filename )
```

Open an existing template bundle file.

Parameters

_filename	Name of the file to open.

7.21.3.12 outputXMLToPDF()

```
RolePlayingDB3::Template::outputXMLToPDF (
            pdfobj ,
            heading  = "Sheet" )
```

Output XML to a PDF file.

Parameters

pdfobj	The PDF object to print to.
heading	The heading.

7.21.3.13 print()

```
RolePlayingDB3::Template::print ( )
```

Print (to a PDF file) the current template bundle.

7.21.3.14 rescantemplates()

```
RolePlayingDB3::Template::rescantemplates ( )
```

Rescan the template tree.

7.21.3.15 save()

```
RolePlayingDB3::Template::save ( )
```

Save the current template bundle in the current file.

7.21.3.16 saveas()

```
RolePlayingDB3::Template::saveas (
            _filename  = "" )
```

Save the current template bundle in specified file.

Parameters

_filename	File name to save to.

The documentation for this class was generated from the following file:

- RPGTemplate.tcl

7.22 RolePlayingDB3::XMLContentEditor Class Reference

Public Member Functions

- getElementWidgetById (id)
- getpageno ()
- getlineno ()
- bcmdmethod (id="")
- XMLContentEditor (...)
- ~XMLContentEditor ()
- recreateXML (file)
- outputXMLToPDF (pdfobj, heading="Sheet", curpage=0, curline=1000)

Static Public Member Functions

- static ExtractTagValue (xmlstring, name, default)

7.22.1 Detailed Description

Widget to for editing XML Content. Takes a XML string and builds a widget to edit it.

7.22.2 Constructor & Destructor Documentation

7.22.2.1 XMLContentEditor()

```
RolePlayingDB3::XMLContentEditor::XMLContentEditor (
              ...  )
```

Constructor

Parameters

widgetpath	The widget path.

Parameters

...	Options: • -xml The XML string (Required). • -templatevariable The template variable. • -dirtyvariable The dirty variable. • -basedirectory The base directory. • -filewidgethandler The file widget handler. • -xmlfile The file to save the XML to. • -buttoncommand The button callback. • -isnewobject Flag to indicate that this is a new entity. • -rootcontainer The root container's tag.

7.22.2.2 ~XMLContentEditor()

```
RolePlayingDB3::XMLContentEditor::~XMLContentEditor ( )
```

Destory the XML tree (free up memory).

7.22.3 Member Function Documentation

7.22.3.1 bcmdmethod()

```
RolePlayingDB3::XMLContentEditor::bcmdmethod (
            id  = "" )
```

Button command method

Parameters

id	Button id

7.22.3.2 ExtractTagValue()

```
static RolePlayingDB3::XMLContentEditor::ExtractTagValue (
            xmlstring ,
            name ,
            default  )  [static]
```

Fetch the value (character data) of a given Field by name in a XML String.

Parameters

xmlstring	The XML string.
name	The name.
default	The value to return if the named field is not found.

7.22.3.3 getElementWidgetById()

```
RolePlayingDB3::XMLContentEditor::getElementWidgetById (
            id  )
```

Get an element's widget, given the element's id.

Parameters

id	the element id.

Returns

a widget or {}.

7.22.3.4 getlineno()

```
RolePlayingDB3::XMLContentEditor::getlineno ( )
```

Return the line number.

Returns

the line number.

7.22.3.5 getpageno()

```
RolePlayingDB3::XMLContentEditor::getpageno ( )
```

Return page number

Returns

the page number.

7.22.3.6 outputXMLToPDF()

```
RolePlayingDB3::XMLContentEditor::outputXMLToPDF (
            pdfobj ,
            heading  = "Sheet",
            curpage  = 0,
            curline  = 1000 )
```

Output the XML tree as PDF to a pdfobject.

Parameters

pdfobj	The pdfobject to write to.
heading	The main heading.
curpage	The current page number.
curline	The current line number.

7.22.3.7 recreateXML()

```
RolePlayingDB3::XMLContentEditor::recreateXML (
            file  )
```

Regenerate the XML string and save it to a file.

Parameters

file	The name of the file to write to.

The documentation for this class was generated from the following file:

- RPGUtilities.tcl

Index

_
    gettext, 15
_createEditDialog
    RolePlayingDB3::MapEditor, 41
_m
    gettext, 15
_mx
    gettext, 16
∼ParseXML
    ParseXML, 47
∼RPGDirent
    vfs::rpg::RPGDirent, 51
∼RPGFileSystem
    vfs::rpg::RPGFileSystem, 55
∼SimpleDOMElement
    SimpleDOMElement, 64
∼XMLContentEditor
    RolePlayingDB3::XMLContentEditor, 82

add
    DynamicHelp, 24
AddNewDirent
    vfs::rpg::RPGDirent, 51
addchild
    SimpleDOMElement, 65
appenddata
    SimpleDOMElement, 65
attribute
    SimpleDOMElement, 65

bcmdmethod
    RolePlayingDB3::XMLContentEditor, 82
bind
    RolePlayingDB3::LabeledScrolledText, 32
bindcanvas
    RolePlayingDB3::ScrolledCanvas, 56
binditem
    RolePlayingDB3::LabeledDirTree, 30
bitmap
    IconBitmap, 27

checksave
    RolePlayingDB3::LevelEditor, 33
    RolePlayingDB3::SpaceEditor, 71
children
    SimpleDOMElement, 66
close
    RolePlayingDB3::LevelEditor, 33
    RolePlayingDB3::MapEditor, 41
    RolePlayingDB3::SheetEdit, 59
    RolePlayingDB3::SpaceEditor, 71
    RolePlayingDB3::Template, 77
colorchanged
    RolePlayingDB3::SpaceEditor, 72
ConfigurationType
    ReadConfiguration, 16
configure
    DynamicHelp, 24

data
    SimpleDOMElement, 66
delete
    DynamicHelp, 25
Dialog, 23
display
    SimpleDOMElement, 66
displayTree
    ParseXML, 47
drawonespace
    RolePlayingDB3::LevelEditor, 34
drawspace
    RolePlayingDB3::LevelEditor, 34
    RolePlayingDB3::MapEditor, 41
DynamicHelp, 24
    add, 24
    configure, 24
    delete, 25
    register, 25
DynamicHelp_include
    TclCommon, 13

edit
    RolePlayingDB3::MapEditor, 42
    RolePlayingDB3::SheetEdit, 60
    RolePlayingDB3::Template, 77
ExtractTagValue
    RolePlayingDB3::XMLContentEditor, 82

fail
    vfs::rpg, 21

GeneratePrintDialog

RolePlayingDB3, 19
GetDirents
vfs::rpg::RPGDirent, 51
getElementWidgetById
RolePlayingDB3::XMLContentEditor, 83
getElementsById
SimpleDOMElement, 66
getElementsByTagName
SimpleDOMElement, 67
getParent
SimpleDOMElement, 67
getcolor
RolePlayingDB3::SpaceEditor, 72
getfile
RolePlayingDB3::SheetEdit, 60
RolePlayingDB3::Template, 77
getimage
RolePlayingDB3::Graphic, 26
getlineno
RolePlayingDB3::XMLContentEditor, 83
getpageno
RolePlayingDB3::XMLContentEditor, 83
getspacefile
RolePlayingDB3::SpaceEditor, 72
gettext, 15
_, 15
_m, 15
_mx, 16
Graphic
RolePlayingDB3::Graphic, 26

IconBitmap, 27
bitmap, 27
IconImage, 28
image, 28
image
IconImage, 28
insert
RolePlayingDB3::ScrolledList, 57
isChild
SimpleDOMElement, 68
IsEven
ReadConfiguration, 17
itemcget
RolePlayingDB3::LabeledDirTree, 30
RolePlayingDB3::ScrolledList, 58
items
RolePlayingDB3::ScrolledList, 58

LabeledDirTree
RolePlayingDB3::LabeledDirTree, 29
LabeledScrolledText
RolePlayingDB3::LabeledScrolledText, 31
length
SimpleDOMElement, 68
LevelDirOption
RolePlayingDB3, 19
LevelEditor
RolePlayingDB3::LevelEditor, 33
LevelEditorOption
RolePlayingDB3, 19
LookupMatch
vfs::rpg::RPGDirent, 52
LookupName
vfs::rpg::RPGDirent, 52

MapBundleMountPoint
RolePlayingDB3, 19
MapEditor
RolePlayingDB3::MapEditor, 40
MapEditorOption
RolePlayingDB3, 20
Mount
vfs::rpg, 21

new
RolePlayingDB3::LevelEditor, 35
RolePlayingDB3::MapEditor, 42
RolePlayingDB3::SheetEdit, 60, 61
RolePlayingDB3::SpaceEditor, 72, 73
RolePlayingDB3::Template, 77, 78
newPDFPage
RolePlayingDB3::Template, 78
NumDirents
vfs::rpg::RPGDirent, 53

open
RolePlayingDB3::LevelEditor, 35
RolePlayingDB3::MapEditor, 42
RolePlayingDB3::SheetEdit, 61
RolePlayingDB3::SpaceEditor, 73
RolePlayingDB3::Template, 78, 79
OpenFilename
RolePlayingDB3, 20
openfile
RolePlayingDB3::MapEditor, 43
RolePlayingDB3::SheetEdit, 61
RolePlayingDB3::Template, 79
opennew
RolePlayingDB3::MapEditor, 43
RolePlayingDB3::SheetEdit, 62
RolePlayingDB3::Template, 79
openold
RolePlayingDB3::MapEditor, 43
RolePlayingDB3::SheetEdit, 62
RolePlayingDB3::Template, 79
outputXMLToPDF
RolePlayingDB3::Template, 79
RolePlayingDB3::XMLContentEditor, 84

ParseXML, 46
~ParseXML, 47
displayTree, 47
ParseXML, 46
TopContainer, 47
validate, 48
print
RolePlayingDB3::LevelEditor, 36
RolePlayingDB3::MapEditor, 44
RolePlayingDB3::SheetEdit, 62
RolePlayingDB3::SpaceEditor, 74
RolePlayingDB3::Template, 80
printLevel
RolePlayingDB3::LevelEditor, 36
printSpace
RolePlayingDB3::SpaceEditor, 74
printprogress_end
RolePlayingDB3::PrintDialog, 49
printprogress_setpageno
RolePlayingDB3::PrintDialog, 49
printprogress_start
RolePlayingDB3::PrintDialog, 49

RPGDirent
vfs::rpg::RPGDirent, 50
RPGFileSystem
vfs::rpg::RPGFileSystem, 54
RPGSupport, 14
ReadConfiguration, 16
ConfigurationType, 16
IsEven, 17
ReadConfiguration, 17
WriteConfiguration, 18
recreateXML
RolePlayingDB3::LevelEditor, 37
RolePlayingDB3::MapEditor, 44
RolePlayingDB3::SheetEdit, 62
RolePlayingDB3::SpaceEditor, 75
RolePlayingDB3::XMLContentEditor, 84
redrawdirtree
RolePlayingDB3::LabeledDirTree, 30
redrawspace
RolePlayingDB3::SpaceEditor, 75
register
DynamicHelp, 25
removeChild
SimpleDOMElement, 68
RemoveDirent
vfs::rpg::RPGDirent, 53
removeeditor
RolePlayingDB3::LevelEditor, 37
replaceChildren
SimpleDOMElement, 69
rescalelevelmap
RolePlayingDB3::LevelEditor, 37
rescantemplates
RolePlayingDB3::Template, 80
RolePlayingDB3, 18
GeneratePrintDialog, 19
LevelDirOption, 19
LevelEditorOption, 19
MapBundleMountPoint, 19
MapEditorOption, 20
OpenFilename, 20
SaveFilename, 20
SpaceFileOption, 20
RolePlayingDB3::Graphic, 26
getimage, 26
Graphic, 26
RolePlayingDB3::LabeledDirTree, 28
binditem, 30
itemcget, 30
LabeledDirTree, 29
redrawdirtree, 30
RolePlayingDB3::LabeledScrolledText, 31
bind, 32
LabeledScrolledText, 31
RolePlayingDB3::LevelEditor, 32
checksave, 33
close, 33
drawonespace, 34
drawspace, 34
LevelEditor, 33
new, 35
open, 35
print, 36
printLevel, 36
recreateXML, 37
removeeditor, 37
rescalelevelmap, 37
save, 37
saveas, 37
setspaceposition, 38
updatelevelmap, 38
updatelevelmapcolor, 38
updateleveltree, 39
updatemediatree, 39
updatespacetree, 39
RolePlayingDB3::MapEditor, 40
_createEditDialog, 41
close, 41
drawspace, 41
edit, 42
MapEditor, 40
new, 42
open, 42
openfile, 43
opennew, 43

openold, 43
print, 44
recreateXML, 44
save, 44
saveas, 44
scaleXY, 44
unscaleXY, 45
updateleveltree, 45
updatemediatree, 45
RolePlayingDB3::PrintDialog, 48
printprogress_end, 49
printprogress_setpageno, 49
printprogress_start, 49
RolePlayingDB3::ScrolledCanvas, 55
bindcanvas, 56
ScrolledCanvas, 55
RolePlayingDB3::ScrolledList, 56
insert, 57
itemcget, 58
items, 58
ScrolledList, 57
RolePlayingDB3::SheetEdit, 58
close, 59
edit, 60
getfile, 60
new, 60, 61
open, 61
openfile, 61
opennew, 62
openold, 62
print, 62
recreateXML, 62
save, 62
saveas, 63
SheetEdit, 59
RolePlayingDB3::SpaceEditor, 70
checksave, 71
close, 71
colorchanged, 72
getcolor, 72
getspacefile, 72
new, 72, 73
open, 73
print, 74
printSpace, 74
recreateXML, 75
redrawspace, 75
save, 75
saveas, 75
setdirty, 75
SpaceEditor, 71
RolePlayingDB3::Template, 76
close, 77
edit, 77
getfile, 77
new, 77, 78
newPDFPage, 78
open, 78, 79
openfile, 79
opennew, 79
openold, 79
outputXMLToPDF, 79
print, 80
rescantemplates, 80
save, 80
saveas, 80
Template, 76
RolePlayingDB3::XMLContentEditor, 81
∼XMLContentEditor, 82
bcmdmethod, 82
ExtractTagValue, 82
getElementWidgetById, 83
getlineno, 83
getpageno, 83
outputXMLToPDF, 84
recreateXML, 84
XMLContentEditor, 81
RolePlayingDB3::chroot_chooseDirectory, 23
RolePlayingDB3::chroot_getFile, 23

save
RolePlayingDB3::LevelEditor, 37
RolePlayingDB3::MapEditor, 44
RolePlayingDB3::SheetEdit, 62
RolePlayingDB3::SpaceEditor, 75
RolePlayingDB3::Template, 80
SaveFilename
RolePlayingDB3, 20
saveas
RolePlayingDB3::LevelEditor, 37
RolePlayingDB3::MapEditor, 44
RolePlayingDB3::SheetEdit, 63
RolePlayingDB3::SpaceEditor, 75
RolePlayingDB3::Template, 80
scaleXY
RolePlayingDB3::MapEditor, 44
ScrolledCanvas
RolePlayingDB3::ScrolledCanvas, 55
ScrolledList
RolePlayingDB3::ScrolledList, 57
setAttribute
SimpleDOMElement, 69
setdata
SimpleDOMElement, 69
setdirty
RolePlayingDB3::SpaceEditor, 75
setspaceposition
RolePlayingDB3::LevelEditor, 38

SheetEdit
RolePlayingDB3::SheetEdit, 59
SimpleDOMElement, 63
~SimpleDOMElement, 64
addchild, 65
appenddata, 65
attribute, 65
children, 66
data, 66
display, 66
getElementsById, 66
getElementsByTagName, 67
getParent, 67
isChild, 68
length, 68
removeChild, 68
replaceChildren, 69
setAttribute, 69
setdata, 69
SimpleDOMElement, 64
validate, 70
SpaceEditor
RolePlayingDB3::SpaceEditor, 71
SpaceFileOption
RolePlayingDB3, 20

TclCommon, 13
DynamicHelp_include, 13
Template
RolePlayingDB3::Template, 76
TopContainer
ParseXML, 47

unscaleXY
RolePlayingDB3::MapEditor, 45
updatelevelmap
RolePlayingDB3::LevelEditor, 38
updatelevelmapcolor
RolePlayingDB3::LevelEditor, 38
updateleveltree
RolePlayingDB3::LevelEditor, 39
RolePlayingDB3::MapEditor, 45
updatemediatree
RolePlayingDB3::LevelEditor, 39
RolePlayingDB3::MapEditor, 45
updatespacetree
RolePlayingDB3::LevelEditor, 39

validate
ParseXML, 48
SimpleDOMElement, 70
vfs::rpg, 21
fail, 21
Mount, 21
vfs::rpg::RPGDirent, 49
~RPGDirent, 51
AddNewDirent, 51
GetDirents, 51
LookupMatch, 52
LookupName, 52
NumDirents, 53
RPGDirent, 50
RemoveDirent, 53
vfs::rpg::RPGFileSystem, 54
~RPGFileSystem, 55
RPGFileSystem, 54

WriteConfiguration
ReadConfiguration, 18

XMLContentEditor
RolePlayingDB3::XMLContentEditor, 81