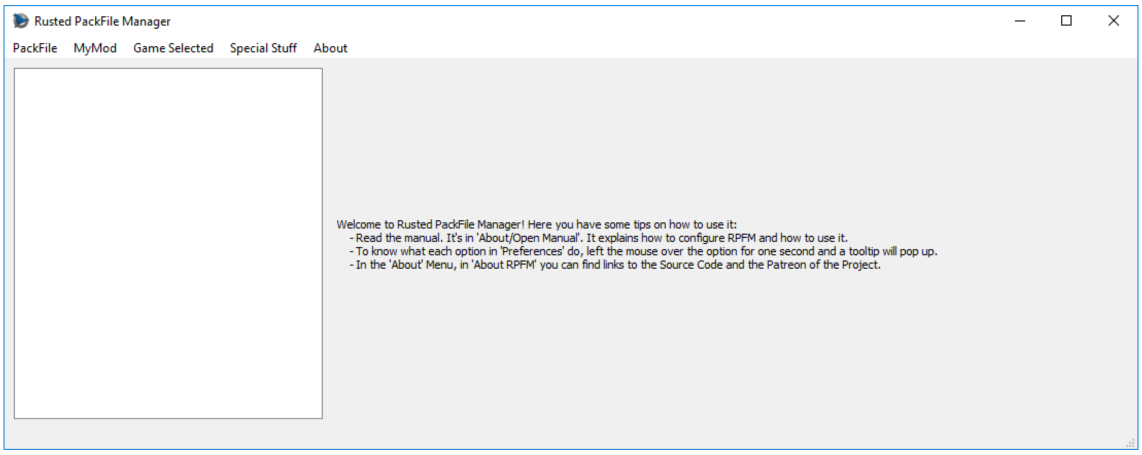
Rusted PackFile Manager

By *Ismael Gutiérrez González*, A.K.A "***Frodo45127***"

Version 1.1, available here: https://github.com/Frodo45127/rpfm

Introduction

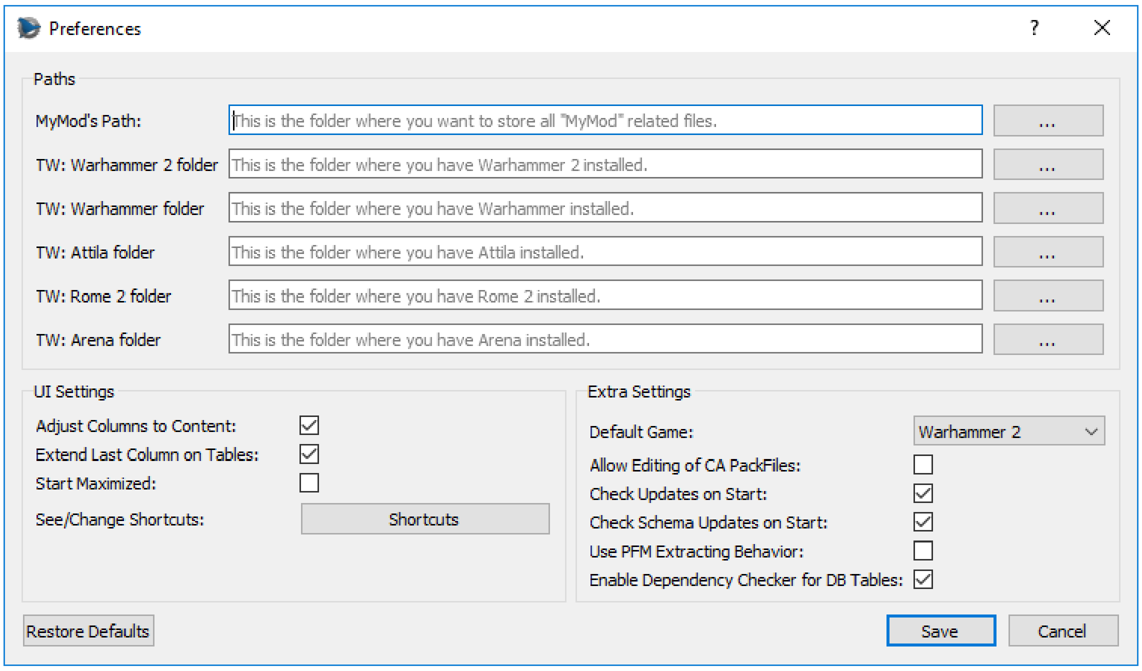


Welcome to Rusted PackFile Manager! Here you have some tips on how to use it:
- Read the manual. It's in 'About/Open Manual'. It explains how to configure RPFM and how to use it.
- To know what each option in 'Preferences' do, left the mouse over the option for one second and a tooltip will pop up.
- In the 'About' Menu, in 'About RPFM' you can find links to the Source Code and the Patreon of the Project.

This is ***Rusted PackFile Manager***, or ***RPFM***, a modding tool for modern Total War Games. If you've used PFM before, you can see it has a similar UI. It's done that way to make it easier to use for modders coming from PFM. We are going to do the initial configuration first, and then we are going to take a look at the different features RPFM has to offer.

Remember that, since version 1.0, this manual should always be in RPFM's folder, or accessible from "***About/Open Manual***" in RPFM's Menu Bar, in case you want to check it later.

Initial Configuration



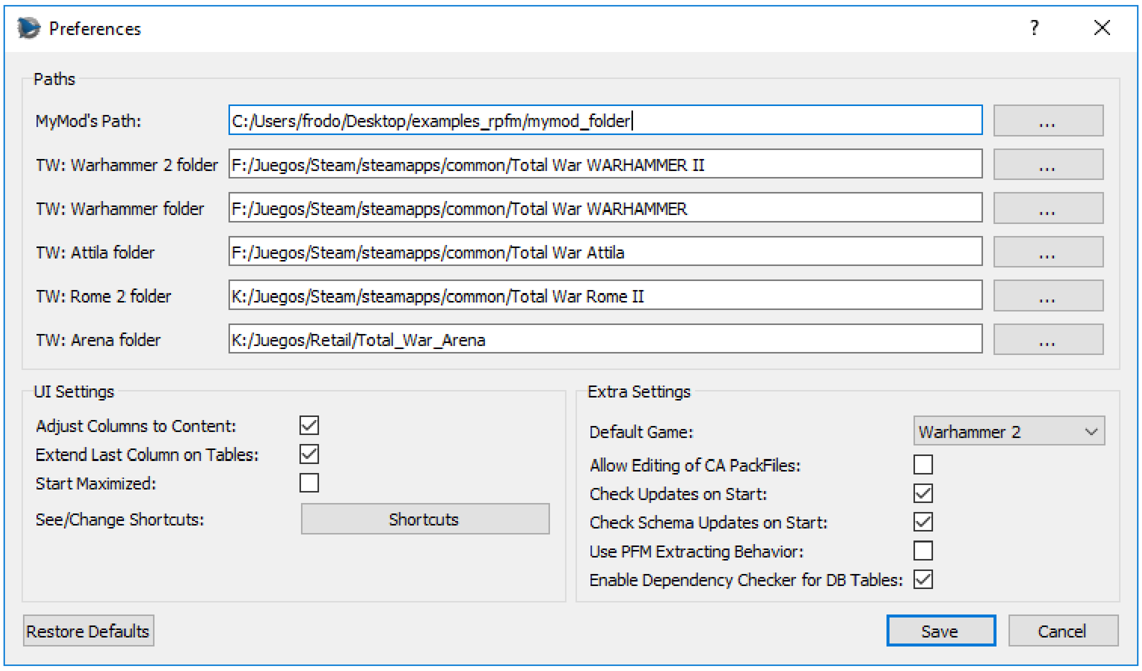
MyMod's Path:	This is the folder where you want to store all "MyMod" related files.
TW: Warhammer 2 folder	This is the folder where you have Warhammer 2 installed.
TW: Warhammer folder	This is the folder where you have Warhammer installed.
TW: Attila folder	This is the folder where you have Attila installed.
TW: Rome 2 folder	This is the folder where you have Rome 2 installed.
TW: Arena folder	This is the folder where you have Arena installed.

Adjust Columns to Content:	✓
Extend Last Column on Tables:	✓
Start Maximized:	☐
See/Change Shortcuts:	Shortcuts

Default Game:	Warhammer 2
Allow Editing of CA PackFiles:	☐
Check Updates on Start:	✓
Check Schema Updates on Start:	✓
Use PFM Extracting Behavior:	☐
Enable Dependency Checker for DB Tables:	✓

After the first start, we need to go to ***"PackFile/Preferences"***, and configure the stuff we want. First, the ***"MyMod"*** Path. This is the path where ***"MyMods"*** and their data will be stored. We'll see later in the tutorial what this *"MyMods"* thing is. For this tutorial, we'll pick an empty folder. Then, the games paths. These are the folders where each game is installed. We'll fill the ones for the games we have installed.

In the end, it should look something like this:



Preferences

Paths

MyMod's Path:	C:/Users/frodo/Desktop/examples_rpfm/mymod_folder
TW: Warhammer 2 folder	F:/Juegos/Steam/steamapps/common/Total War WARHAMMER II
TW: Warhammer folder	F:/Juegos/Steam/steamapps/common/Total War WARHAMMER
TW: Attila folder	F:/Juegos/Steam/steamapps/common/Total War Attila
TW: Rome 2 folder	K:/Juegos/Steam/steamapps/common/Total War Rome II
TW: Arena folder	K:/Juegos/Retail/Total_War_Arena

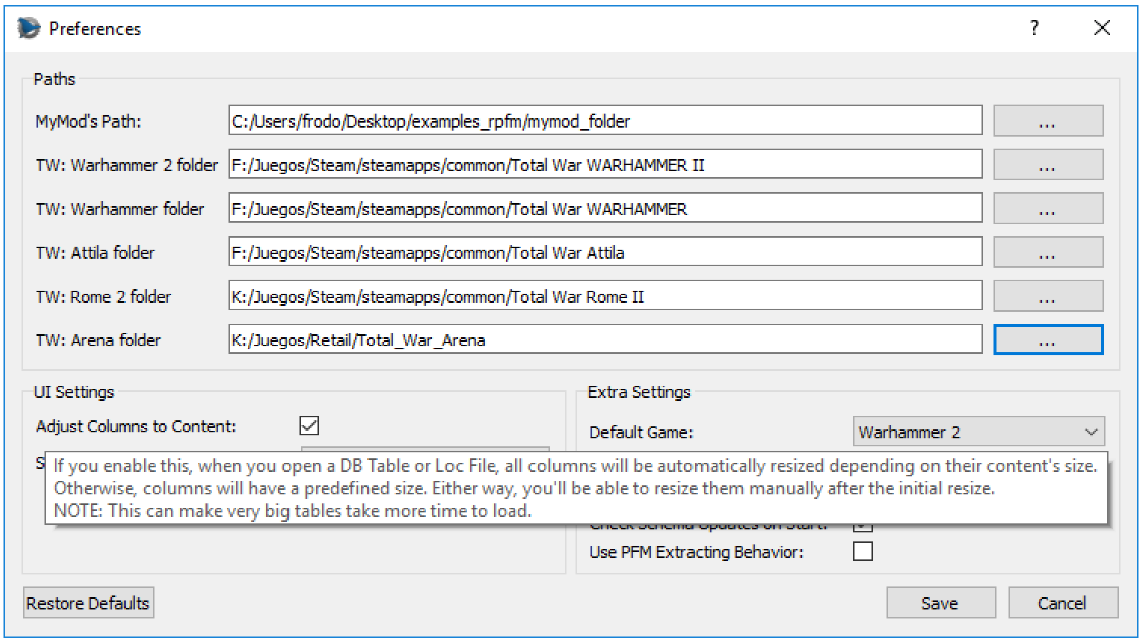
UI Settings

Adjust Columns to Content:	✓
Extend Last Column on Tables:	✓
Start Maximized:	☐
See/Change Shortcuts:	Shortcuts

Extra Settings

Default Game:	Warhammer 2
Allow Editing of CA PackFiles:	☐
Check Updates on Start:	✓
Check Schema Updates on Start:	✓
Use PFM Extracting Behavior:	☐
Enable Dependency Checker for DB Tables:	✓

Now we choose our "***Default Game***". RPFM uses a ***"Game Selected"*** setting to configure certain parts of the program to work with a game or another. For example, it changes the way the mods are saved, the default folder to save them, the schema used for the tables,.... Here you can set the game that'll be selected by default when you open the program. We'll leave it in ***"Warhammer 2"*** for this tutorial. Next, all these checkboxes. You can get an explanation about what they do just by hovering them with the mouse, like this.



Preferences

Paths

MyMod's Path:	C:/Users/frodo/Desktop/examples_rpfm/mymod_folder
TW: Warhammer 2 folder	F:/Juegos/Steam/steamapps/common/Total War WARHAMMER II
TW: Warhammer folder	F:/Juegos/Steam/steamapps/common/Total War WARHAMMER
TW: Attila folder	F:/Juegos/Steam/steamapps/common/Total War Attila
TW: Rome 2 folder	K:/Juegos/Steam/steamapps/common/Total War Rome II
TW: Arena folder	K:/Juegos/Retail/Total_War_Arena

UI Settings

Adjust Columns to Content: ✓

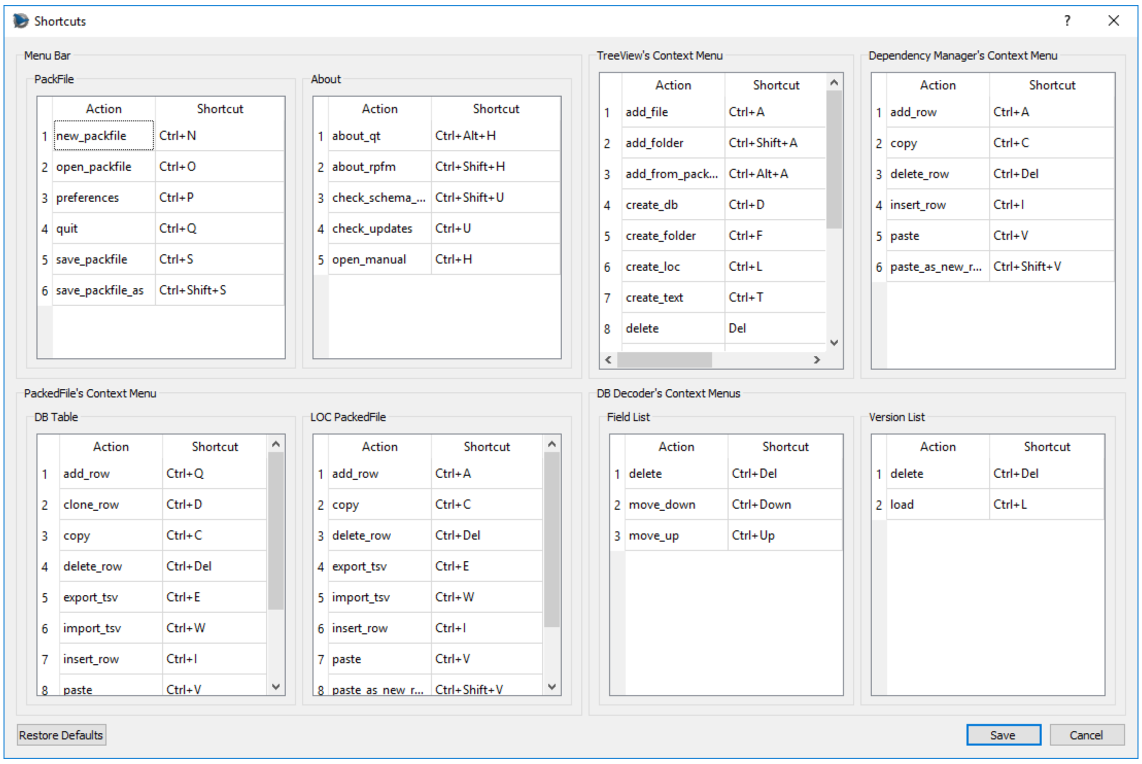
Extra Settings

Default Game: Warhammer 2

Use PFM Extracting Behavior: ☐

If you enable this, when you open a DB Table or Loc File, all columns will be automatically resized depending on their content's size. Otherwise, columns will have a predefined size. Either way, you'll be able to resize them manually after the initial resize.
NOTE: This can make very big tables take more time to load.

And finally, the "***Shortcuts***" button. Hitting it will open the "***Shortcuts***" window, were you can see and edit all the shortcuts currently used by RPFM.



Menu Bar

PackFile

	Action	Shortcut
1	new_packfile	Ctrl+N
2	open_packfile	Ctrl+O
3	preferences	Ctrl+P
4	quit	Ctrl+Q
5	save_packfile	Ctrl+S
6	save_packfile_as	Ctrl+Shift+S

About

	Action	Shortcut
1	about_qt	Ctrl+Alt+H
2	about_rpfm	Ctrl+Shift+H
3	check_schema_...	Ctrl+Shift+U
4	check_updates	Ctrl+U
5	open_manual	Ctrl+H

TreeView's Context Menu

	Action	Shortcut
1	add_file	Ctrl+A
2	add_folder	Ctrl+Shift+A
3	add_from_pack...	Ctrl+Alt+A
4	create_db	Ctrl+D
5	create_folder	Ctrl+F
6	create_loc	Ctrl+L
7	create_text	Ctrl+T
8	delete	Del

Dependency Manager's Context Menu

	Action	Shortcut
1	add_row	Ctrl+A
2	copy	Ctrl+C
3	delete_row	Ctrl+Del
4	insert_row	Ctrl+I
5	paste	Ctrl+V
6	paste_as_new_r...	Ctrl+Shift+V

PackedFile's Context Menu

DB Table

	Action	Shortcut
1	add_row	Ctrl+Q
2	clone_row	Ctrl+D
3	copy	Ctrl+C
4	delete_row	Ctrl+Del
5	export_tsv	Ctrl+E
6	import_tsv	Ctrl+W
7	insert_row	Ctrl+I
8	paste	Ctrl+V

LOC PackedFile

	Action	Shortcut
1	add_row	Ctrl+A
2	copy	Ctrl+C
3	delete_row	Ctrl+Del
4	export_tsv	Ctrl+E
5	import_tsv	Ctrl+W
6	insert_row	Ctrl+I
7	paste	Ctrl+V
8	paste_as_new_r...	Ctrl+Shift+V

DB Decoder's Context Menus

Field List

	Action	Shortcut
1	delete	Ctrl+Del
2	move_down	Ctrl+Down
3	move_up	Ctrl+Up

Version List

	Action	Shortcut
1	delete	Ctrl+Del
2	load	Ctrl+L

Just keep in mind that some of **<u>the shortcuts are applied when the program starts</u>**, so you'll have to close and re-open RPFM for the changes to take effect.

When you're done with the settings, just hit "***Save***". You can restore them to the defaults with the button of the left (same for the shortcuts with their "***Restore Defaults***" button).

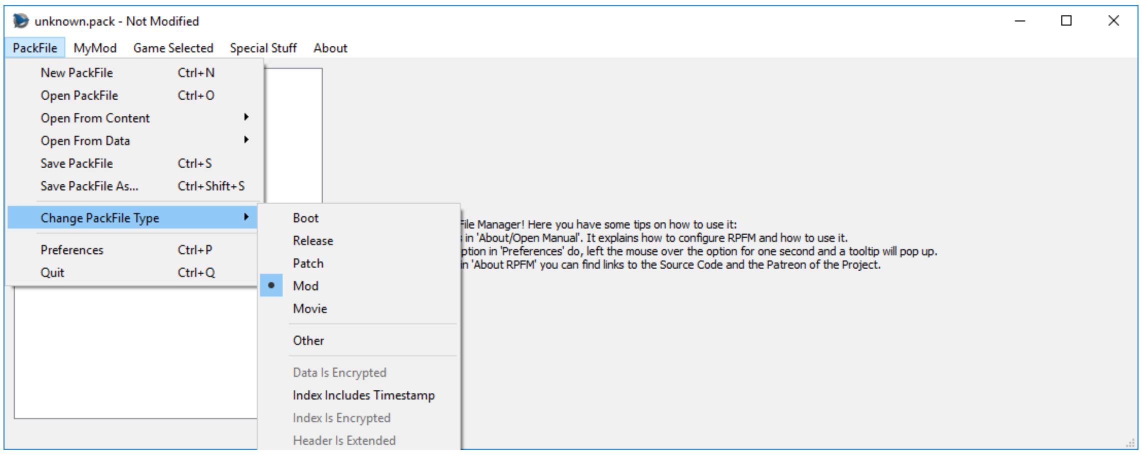
With that, we have completed the initial configuration. Starting on version 1.0, new updates should continue to work with the same settings/shortcuts (as long as new big things aren't added), updating them automatically in case a new setting/shortcut is introduced, storing the saved configurations in the files "***settings.json***" and "***shortcuts.json***", in RPFM's folder.

So now that we're done configuring RPFM, let's take a look at the features it has to offer.

Menu Bar Features

Any action in the Menu Bar has a little tip that'll show up in the Status Bar when selecting it, just in case you want to know what them do. So, we are going to explain here the…. Not very common features, menu by menu.

PackFile Menu



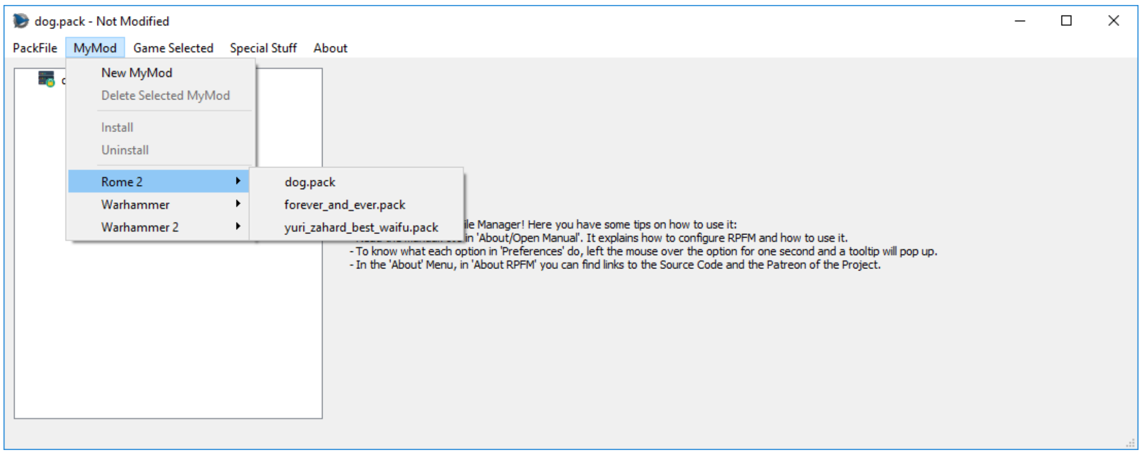
ile Manager! Here you have some tips on how to use it:
in 'About/Open Manual'. It explains how to configure RPFM and how to use it.
ption in 'Preferences' do, left the mouse over the option for one second and a tooltip will pop up.
n 'About RPFM' you can find links to the Source Code and the Patreon of the Project.

Here, we can find the "Basic" actions: ***New***, ***Open***, ***Save***, ***Save As***, ***Preferences*** and ***Quit***. Then we have "***Change PackFile Type***", where you can select the type of the currently open PackFile, and see some special configurations some PackFiles have, like if certain parts of the PackFile are encrypted. If the path of the selected game is configured in the settings, there will be two new submenus: ***Open From Content*** and ***Open From Data***. These submenus will contain shortcuts to open PackFiles from the /data folder of the game, and from /content, where workshop-downloaded PackFiles are. Keep in mind that the ***Open From Content*** submenu will not appear if there have no mods downloaded/uploaded from the workshop.

For the types, for modding you'll only want to use the "***Mod***" type, and maybe, in very specific situations, the "***Movie***" type. "***Boot***", "***Release***", and "***Patch***" are types used by CA PackFiles, not suitable for modding. "***Other***" is for weird or special PackFiles. Do not use it.

Something to remember is that, by default, RPFM doesn't let you save a PackFile if it's "***Boot***", "***Release***" or "***Patch***". If you still want to do it, enable the "Allow Editing of CA PackFiles" setting. "***Other***" PackFiles and PackFiles with "***Data Is Encrypted***", "***Index Is Encrypted***" or "***Header Is Extended***" cannot be saved in any way.

MyMod's Menu



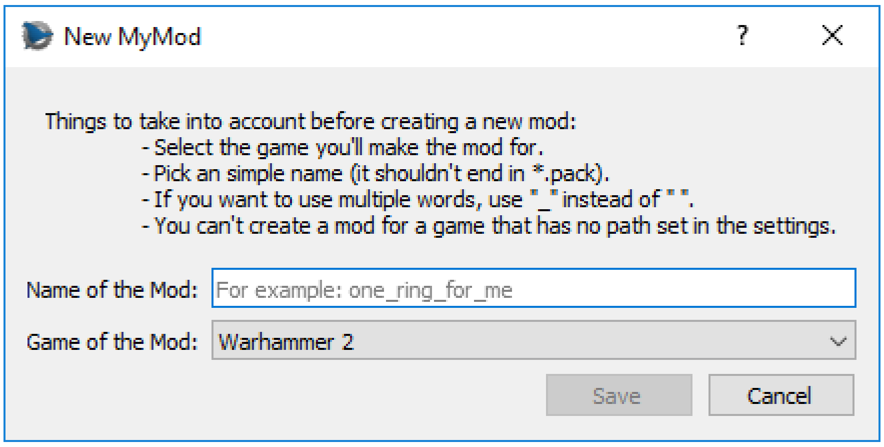
le Manager! Here you have some tips on how to use it:
in 'About/Open Manual'. It explains how to configure RPFM and how to use it.
- To know what each option in 'Preferences' do, left the mouse over the option for one second and a tooltip will pop up.
- In the 'About' Menu, in 'About RPFM' you can find links to the Source Code and the Patreon of the Project.

MyMods are a way to keep data organized when creating a mod. The system is almost a 1:1 clone of PFM's MyMods so it should be easy to use for veterans too.

To those new with the concept, each MyMod has **a *PackFile* (the mod) and a *folder***. Each time you extract something from the PackFile, it'll be automatically extracted in his folder, mirroring the structure it has in the PackFile. For example, extracting a table will result in the table being extracted at *"mymod_folder/db/table_name/file"*. Adding Files/Folders from the MyMod folder will also add them mirroring the path they have. For example, adding a file from *"mymod_folder/db/table_name/file"* will add the file in *"PackFile/db/table_name/file"*.

This makes easier to keep track of the mod files, and you can even put that folder under .git, or any other version control system.

To create a MyMod, just hit ***MyMod/New MyMod*** and this dialog will appear:



Things to take into account before creating a new mod:
- Select the game you'll make the mod for.
- Pick an simple name (it shouldn't end in *.pack).
- If you want to use multiple words, use "_" instead of " ".
- You can't create a mod for a game that has no path set in the settings.

Name of the Mod: For example: one_ring_for_me

Game of the Mod: Warhammer 2

Here you can configure the name and game the mod is for. Once you hit save, your new MyMod will be created and opened.

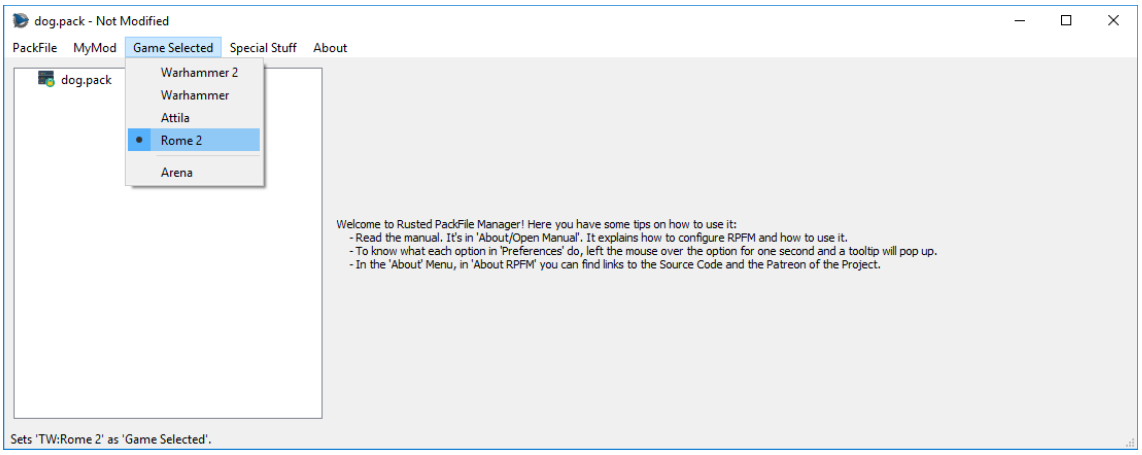
To delete a MyMod, open it and hit ***MyMod/Delete Selected MyMod.*** As this operation will delete the MyMod and the folder, a warning will popup, just in case.

Once you want to test your mod in the game, hit ***MyMod/Install*** and the PackFile will automatically be copied to the /data folder of the game, ready to test. If you make any change in the mod, you'll need to hit install again to update the /data copy of the mod.

If you don't want your mod to show up ingame anymore, hit ***MyMod/Uninstall***. This will remove your mod from the data folder.

Keep in mind that, to execute those last two commands, you need to have your MyMod open. Under them you'll find the list of MyMods you've created, separated by game. Only MyMod PackFiles opened from these submenus or created using ***MyMod/New MyMod*** will enjoy the features like keeping the paths when adding/extracting files. Manually opened MyMods will be treated as regular PackFiles.

Game Selected Menu

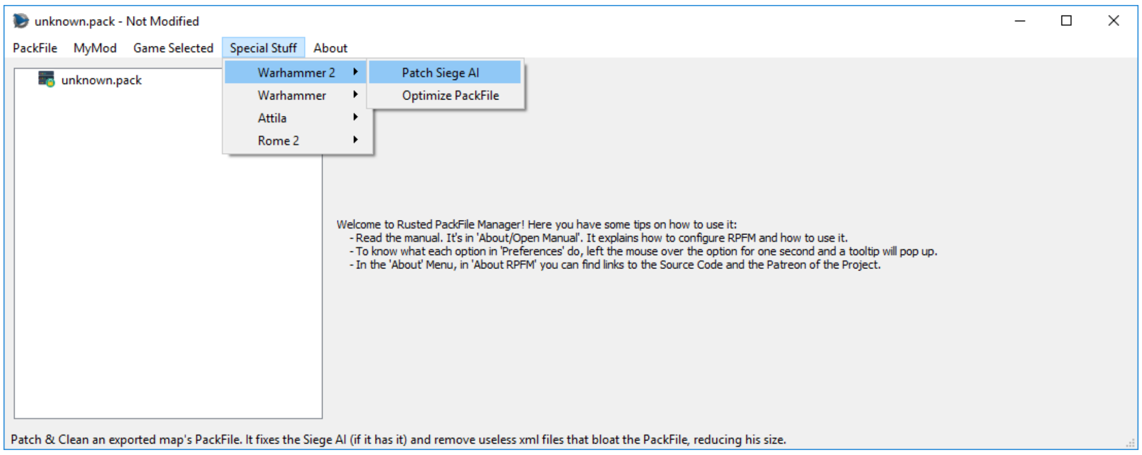


Welcome to Rusted PackFile Manager! Here you have some tips on how to use it:
- Read the manual. It's in 'About/Open Manual'. It explains how to configure RPFM and how to use it.
- To know what each option in 'Preferences' do, left the mouse over the option for one second and a tooltip will pop up.
- In the 'About' Menu, in 'About RPFM' you can find links to the Source Code and the Patreon of the Project.

Sets 'TW:Rome 2' as 'Game Selected'.

This is where you choose the "***Game Selected***" we talked before. When opening PackFiles, RPFM tries to be smart and auto-select a game, but there are some PackFiles that are the same between games (for example, Attila and Warhammer 1 PackFiles are identical), so... just make sure the right game is selected after opening a PackFile, as that affects how many parts of the program work. Also, ***Arena support is READ-ONLY***. You can't save Arena PackFiles.

Special Stuff Menu



Welcome to Rusted PackFile Manager! Here you have some tips on how to use it:
- Read the manual. It's in 'About/Open Manual'. It explains how to configure RPFM and how to use it.
- To know what each option in 'Preferences' do, left the mouse over the option for one second and a tooltip will pop up.
- In the 'About' Menu, in 'About RPFM' you can find links to the Source Code and the Patreon of the Project.

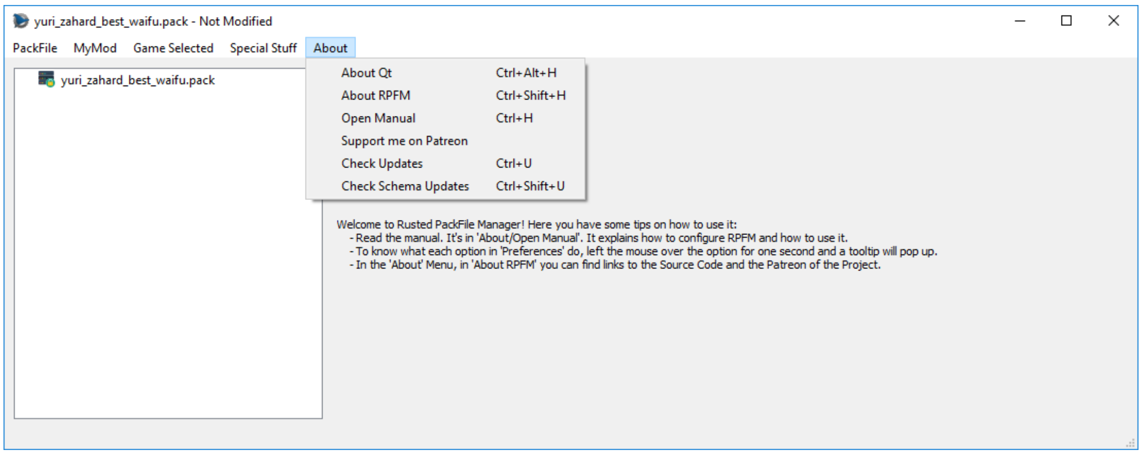
Patch & Clean an exported map's PackFile. It fixes the Siege AI (if it has it) and remove useless xml files that bloat the PackFile, reducing his size.

This has special features implemented for specific games. Here we have "***Patch SiegeAI***", used in Warhammer 1&2 for creating siege maps that the AI can handle.

Also, we have "***Optimize PackFile***", that reduces the size of your PackFile by "*cleaning*" your tables from data that's the same as the one present in the game. For example, if you have a table where all rows but one are exactly the same as the ones in vanilla tables and another table that's a 1:1 copy of a vanilla table without changes, RPFM remove all the rows but the one you changed from the first table, and it'll remove the second table. This is meant to ***improve compatibility with other mods***, and to reduce the size of the PackFile.

This optimization is going to be expanded in the future to Loc PackedFiles and other "useless" files, but for now it only affects DB Tables.

About Menu

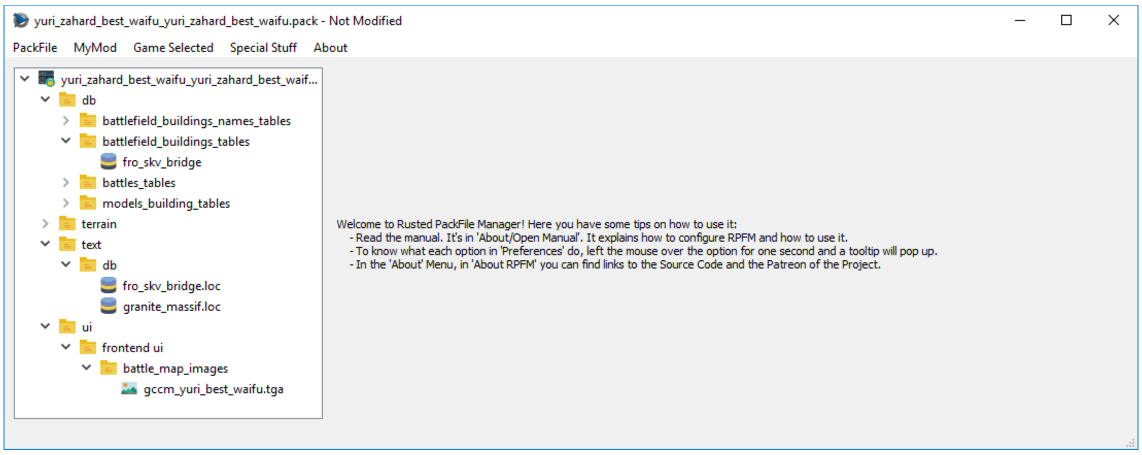


Welcome to Rusted PackFile Manager! Here you have some tips on how to use it:
- Read the manual. It's in 'About/Open Manual'. It explains how to configure RPFM and how to use it.
- To know what each option in 'Preferences' do, left the mouse over the option for one second and a tooltip will pop up.
- In the 'About' Menu, in 'About RPFM' you can find links to the Source Code and the Patreon of the Project.

Here you can find info "***About Qt***" (the UI toolkit use to make RPFM), "***About RPFM***", a button to open this manual in an PDF Reader, a link to RPFM's Patreon that I'm using for important releases, and as a kind of dev blog, and the updaters. The "***Check Updates***" checks if there is any update available for RPFM, and it gives you a link to download it if it finds one. And "***Check Schema Updates***" checks for schema updates and downloads and applies them if you wish. The "***Schema***" is what tells RPFM how to decode the tables of the game. It's very common that after an update a few tables change his structure and are no longer decodables. To get them to work again, the schema has to be updated.

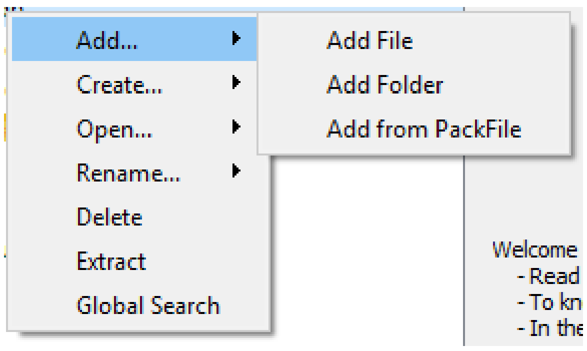
And that's all for the Menu Bar. Now we'll take a look at the TreeView. The TreeView is where all the files inside the PackFile will show up.

PackFile's TreeView



Welcome to Rusted PackFile Manager! Here you have some tips on how to use it:
- Read the manual. It's in 'About/Open Manual'. It explains how to configure RPFM and how to use it.
- To know what each option in 'Preferences' do, left the mouse over the option for one second and a tooltip will pop up.
- In the 'About' Menu, in 'About RPFM' you can find links to the Source Code and the Patreon of the Project.

That thing on the left with folders and stuff is the "***TreeView***". When you Right-Click on any of that stuff, it shows this context menu:



These are the actions we can use to alter the PackFile. Each one of them has a hotkey, in case you're a lazy bastard. These are all the actions in the menu:

- ***Add…/Add File:*** Allows you to add one or more files to the PackFile.
- ***Add…/Add Folder:*** Allows you to add a folder to the PackFile.
- ***Add…/Add from PackFile:*** Allows you to add files or folders from another PackFile. Just, select whatever you want to add, double click it and it'll be added to your PackFile, keeping his path.
- ***Create…/Create Folder:*** Allows you to create an empty folder. Due to how PackFiles work empty folders are not saved so, if you want to keep the folder, add a file to it.
- ***Create …/Create Loc:*** Allows you to create an empty Loc PackedFile.
- ***Create …/Create DB:*** Allows you to create an empty DB Table.
- ***Create …/Create Text:*** Allows you to create an empty text file.
- ***Create …/Mass-Import TSV:*** Allows you to import a bunch of TSV files at once. The system is able to distinguish between DB and Loc TSV files, so you can import all of them at the same time, and RPFM will create all the files needed, in their correct place.
- ***Create …/Mass-Export TSV:*** Allows you to export as TSV every DB Table and Loc PackedFiles inside your PackFile at once.
- ***Open …/Open with Decoder:*** Allows you to open a table in the DB Decoder. Only used to decode new tables, so…. You shouldn't touch this.
- ***Open …/Open Dependency Manager:*** Allows you to open the list of dependencies included in the PackFile. Read a bit more to see what this Dependency Manager thing is about.
- ***Open …/Open with External Program:*** Allows you to open a PackedFile with an external program. Keep in mind that, if you modify the file, changes will NOT BE INCLUDED in the PackedFile itself, but in a file in the TMP folder of your system. If you want to conserve these changes, save that file somewhere, edit it and then add it back to the PackFile.
- ***Rename …/Rename Current:*** Allows you to rename whatever is selected, except the PackFile.
- ***Rename …/Apply Prefix to Selected:*** Allows you to apply a prefix to every file inside the selected folder.
- ***Rename …/Apply Prefix to All:*** Allows you to apply a prefix to every file in the PackFile.
- ***Delete:*** Allows you to delete whatever is selected. If the PackFile is selected, it removes every file from it.
- ***Extract:*** Allows you to extract whatever is selected out of the PackFile.
- ***Global Search:*** Allows you to perform a simple search across every DB Table or Loc PackedFile inside your PackFile, providing you with a filterable list of results.

Additionally, with the shortcuts "**Ctrl++**" and "**Ctrl+-**" you can expand/collapse the entire TreeView. This action is shortcut only, it's not in the Contextual Menu.

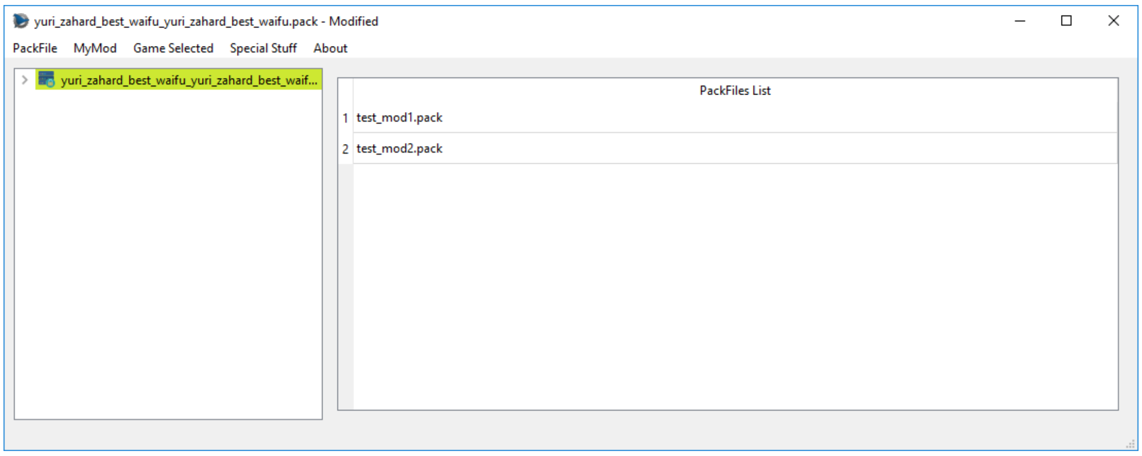
Keep in mind that the availability of these actions depends on what is selected, and on the currently loaded schemas and Dependency PackFile. For example, you can't add anything if you have selected a PackedFile. Also, keep in mind that if there is a "*MyMod*" loaded, some of these actions may work different.

Also, when you add/modify a file, it changes in the TreeView with the following colour code:

- ***Green***: added file.
- ***Yellow***: modified file.
- ***Magenta***: added AND modified file.

This colour code is applied to the parents too, up to the PackFile, so you easily know what you changed since the last time you saved the PackFile.

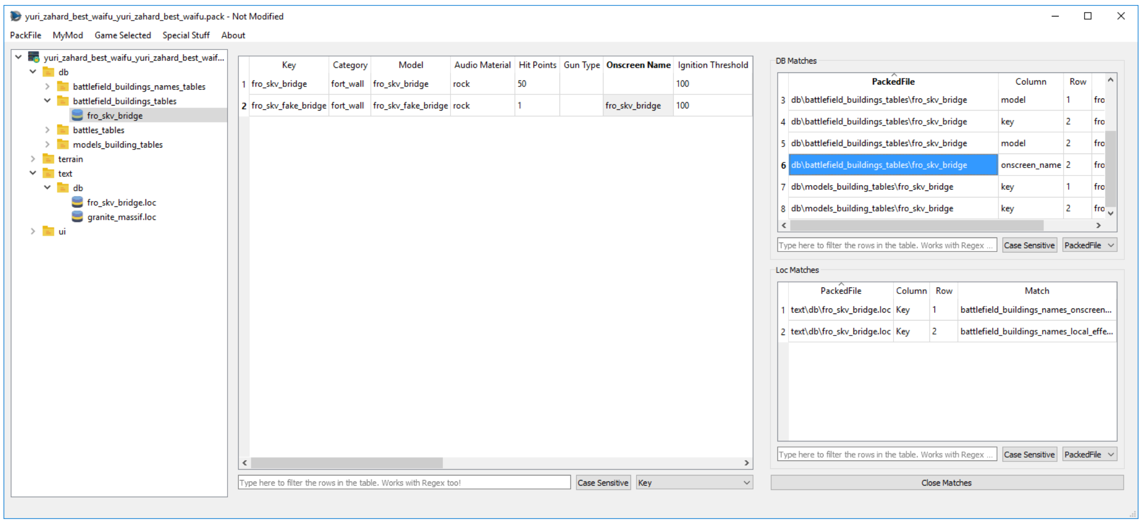
Dependency Manager


yuri_zahard_best_waifu_yuri_zahard_best_waifu.pack - Modified
PackFile
MyMod
Game Selected
Special Stuff
About
yuri_zahard_best_waifu_yuri_zahard_best_waif...
PackFiles List

	PackFiles List
1	test_mod1.pack
2	test_mod2.pack

The ***Dependency Manager*** allows you to modify a special list of PackFiles saved inside your mod's PackFile. When starting the game, the launcher will try to load the PackFiles in this list BEFORE your PackFile. If a PackFile is not found, it'll be ignored. This list can be used to hardcode dependencies into your PackFile. In his Contextual Menu (right-click) you can find some basic commands to manipulate the list, like ***Add Row, Insert Row, Copy, Paste***…

Global Search


yuri_zahard_best_waifu_yuri_zahard_best_waifu.pack - Not Modified
PackFile
MyMod
Game Selected
Special Stuff
About
yuri_zahard_best_waifu_yuri_zahard_best_waif...
db
battlefield_buildings_names_tables
battlefield_buildings_tables
fro_skv_bridge
battles_tables
models_building_tables
terrain
text
db
fro_skv_bridge.loc
granite_massif.loc
ui
Type here to filter the rows in the table. Works with Regex too!
Case Sensitive
Key
DB Matches
Type here to filter the rows in the table. Works with Regex ...
Case Sensitive
PackedFile
Loc Matches
Type here to filter the rows in the table. Works with Regex ...
Case Sensitive
PackedFile
Close Matches

	Key	Category	Model	Audio Material	Hit Points	Gun Type	Onscreen Name	Ignition Threshold
1	fro_skv_bridge	fort_wall	fro_skv_bridge	rock	50			100
2	fro_skv_fake_bridge	fort_wall	fro_skv_fake_bridge	rock	1		fro_skv_bridge	100

DB Matches

	PackedFile	Column	Row	
3	db\battlefield_buildings_tables\fro_skv_bridge	model	1	fro
4	db\battlefield_buildings_tables\fro_skv_bridge	key	2	fro
5	db\battlefield_buildings_tables\fro_skv_bridge	model	2	fro
6	db\battlefield_buildings_tables\fro_skv_bridge	onscreen_name	2	fro
7	db\models_building_tables\fro_skv_bridge	key	1	fro
8	db\models_building_tables\fro_skv_bridge	key	2	fro

Loc Matches

	PackedFile	Column	Row	Match
1	text\db\fro_skv_bridge.loc	Key	1	battlefield_buildings_names_onscreen...
2	text\db\fro_skv_bridge.loc	Key	2	battlefield_buildings_names_local_effe...

Global Search allows you to perform a simple search across every DB Table or Loc PackedFile inside your PackFile, providing you with a filterable list of results in the right of the screen. You can use it from the TreeView's Context Menu, or with the shortcut "**Ctrl+Shift+F**" while the TreeView is focused.

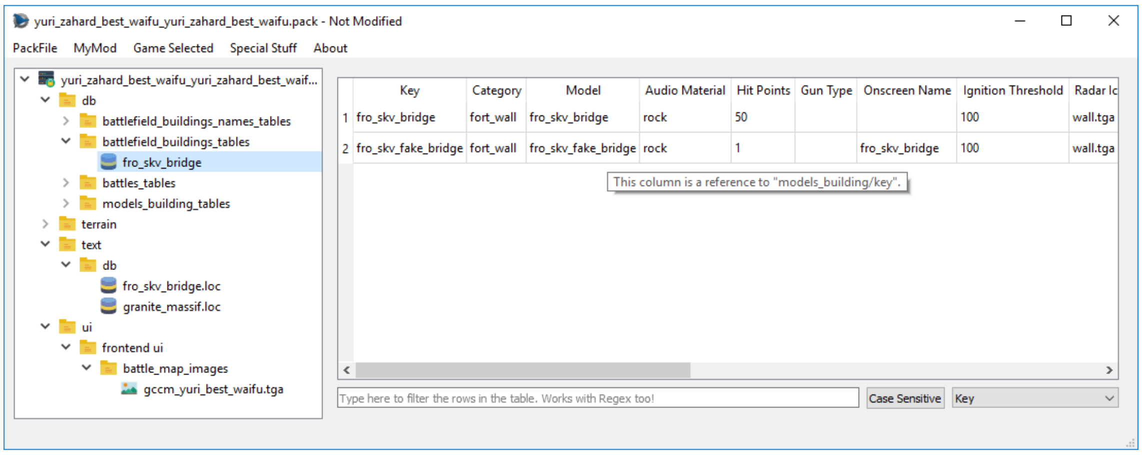
The matches lists on the right of the screen shows every match for your search in **DB Tables** (top table) and **Loc PackedFiles** (bottom table). Both lists ***are filterable*** (with regex support) and contain the path to the PackedFile, Column, Row, and Matched Text. If you double-click on them, their PackedFile **will be open and the match selected**. Also, these lists are updated when you make changes, so if you, for example, remove a match from a table, that match will be removed on-the-fly from the list.

The only inconvenient is that doing this in PackFiles with an enormous amount of tables and a brutal amount of matches will cause RPFM to hang a second after each edit.

PackedFile's Views

Now, time to see the views of each PackedFile's Type RPFM can open.

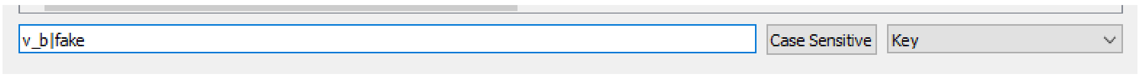
DB PackedFiles


yuri_zahard_best_waifu_yuri_zahard_best_waifu.pack - Not Modified
PackFile
MyMod
Game Selected
Special Stuff
About
yuri_zahard_best_waifu_yuri_zahard_best_waif...
db
battlefield_buildings_names_tables
battlefield_buildings_tables
fro_skv_bridge
battles_tables
models_building_tables
terrain
text
db
fro_skv_bridge.loc
granite_massif.loc
ui
frontend ui
battle_map_images
gccm_yuri_best_waifu.tga
This column is a reference to "models_building/key".
Type here to filter the rows in the table. Works with Regex too!
Case Sensitive
Key

	Key	Category	Model	Audio Material	Hit Points	Gun Type	Onscreen Name	Ignition Threshold	Radar Ic
1	fro_skv_bridge	fort_wall	fro_skv_bridge	rock	50			100	wall.tga
2	fro_skv_fake_bridge	fort_wall	fro_skv_fake_bridge	rock	1		fro_skv_bridge	100	wall.tga

These are encoded tables with most of the "*data*" of the games. This is how a DB Table shows up in RPFM. First, if you hover over any column that references another table, or has a "*description*" in the schema, you can see in the tooltip to what that column references.

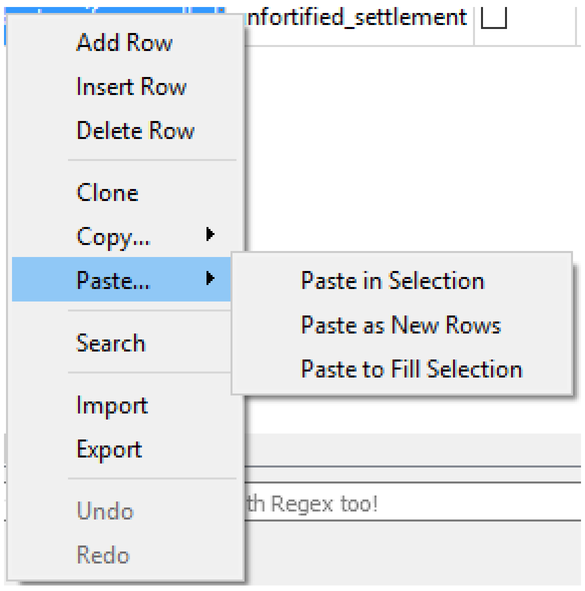
In the bottom of the window you have a ***real-time filter***. Select the column you want to use to filter, if you want it to filter as "*Case Sensitive*", and just write and see how the table gets filtered as you type. It works with Regex too. For example, the following will only show up the rows that contain in their "*Key*" column "***v_b***" or "***fake***":



Here you have a "***Regex Cheatsheet***" in case you want to use more complex filters: https://www.cheatography.com/davechild/cheat-sheets/regular-expressions/

Also, the last filter used before opening another file is saved, so if you reopen a table, the last filter will be automatically re-applied.

Now, with the Right-Click Menu:



These are all the actions available for tables:

- ***Add Row:*** Appends an empty row at the end of the table.
- ***Insert Row:*** Insert an empty row after every row with a selected cell.
- ***Delete Row:*** Uses the computational power of your GPU to mine cryptocurrencies. Joking, it deletes any row with a selected cell.
- ***Clone:*** Creates a duplicate of every row with a selected cell and inserts the duplicate just below the original row.
- ***Copy …/Copy:*** It copies whatever is selected to the Clipboard, in a format compatible with Excel, LibreOffice Calc and others.
- ***Copy …/Copy as LUA Table:*** It copies the entire table as a Lua "Map<String, Vector<data>>" if the table has a key field, or as a series of Vectors if it hasn't, ready to paste it in a script. For scripters.
- ***Paste…/Paste in Selection:*** It tries to paste whatever is in the Clipboard to the selected cells. It does nothing if there are no selected cells, or the clipboard's contents cannot

be pasted into the selected cells without errors. This works by pasting until it ran out of selected cells, or contents to paste.

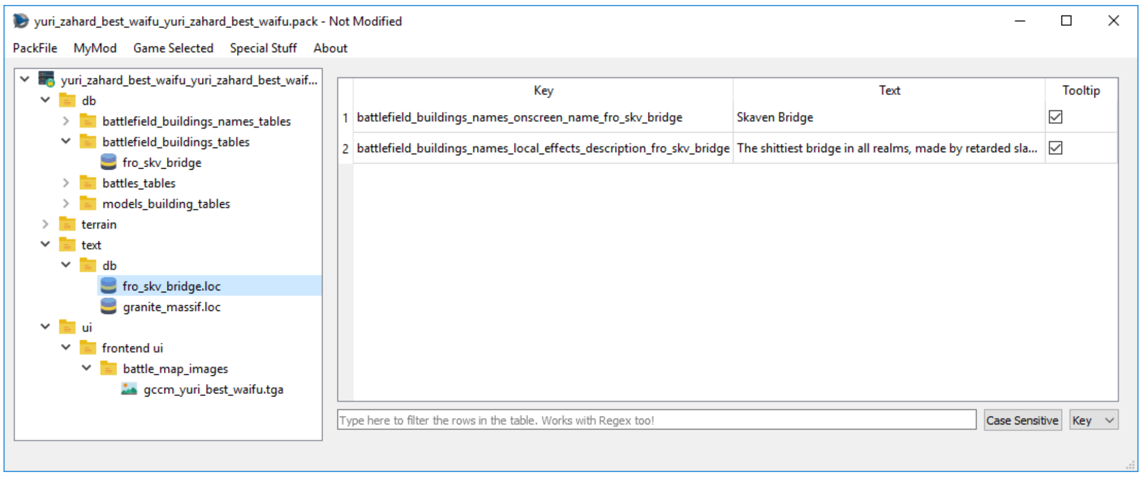
- ***Paste…/Paste as New Rows:*** It tries to paste whatever is in the Clipboard as new rows, appended at the end of the table. It doesn't do anything if the contents of the Clipboard cannot be pasted without errors. In case the contents could be pasted as a "Partial" row, it creates an empty row, and paste what it can paste, leaving the rest of the row empty.
- ***Paste…/Paste to Fill Selection:*** It tries to paste whatever is in the in every selected cell.
- ***Search:*** Open the ***Search&Replace*** panel, that you can use to search any text pattern you want in the table, and replace it if you want. It works in combination with the filter, so you can even do more precise searches combining them!
- ***Import:*** Allows you to import TSV files to the table, overwriting whatever the table currently has. ***IT'S NOT COMPATIBLE WITH PFM TSV FILES***.
- ***Export:*** Allows you to export the table as a TSV File, compatible with Excel, Calc….
- ***Undo:*** Allows you to undo… almost every action done in the table. Even TSV Imports.
- ***Redo:*** Allows you to undo every undo action. This goes deeper into the rabbit hole…

Tables uses the same colour code for cells and rows as the TreeView. And that's more or less what you can do with a DB Table.

Apart of these, the "***Del***" key in DB Tables and PackedFiles acts as an "***Smart Delete***" key. This means depending on what you have selected when you press "Delete" it'll delete:

- **If you have selected random** cells, it'll delete their contents.
- **If you have selected a full row**, it'll remove the row from the table.
- **If you have a combination of both**, it'll delete rows where all cells are selected, and it'll delete the contents of the cells where not all cells in a row are selected. Fancy.

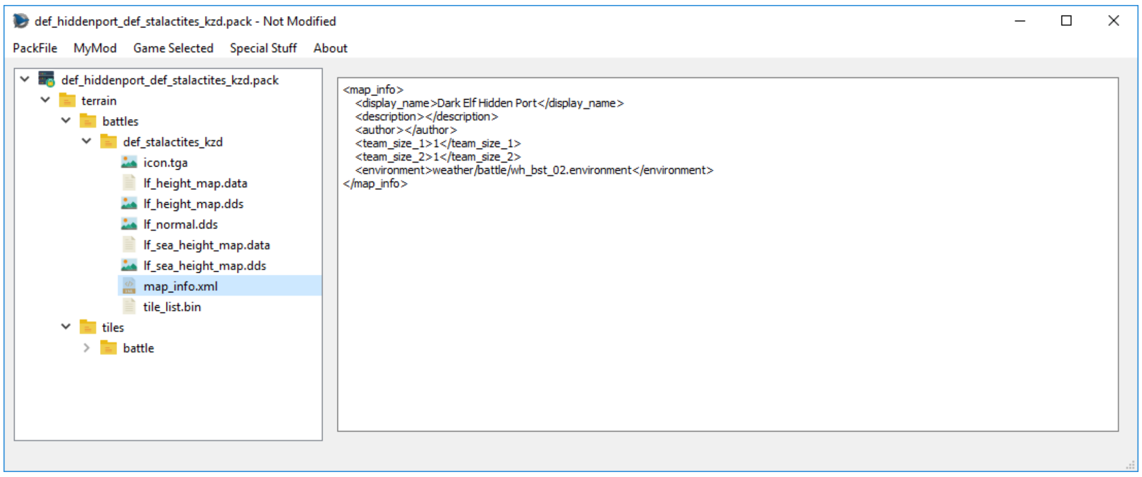
LOC PackedFiles


yuri_zahard_best_waifu_yuri_zahard_best_waifu.pack - Not Modified
PackFile
MyMod
Game Selected
Special Stuff
About
yuri_zahard_best_waifu_yuri_zahard_best_waif...
db
battlefield_buildings_names_tables
battlefield_buildings_tables
fro_skv_bridge
battles_tables
models_building_tables
terrain
text
db
fro_skv_bridge.loc
granite_massif.loc
ui
frontend ui
battle_map_images
gccm_yuri_best_waifu.tga
Type here to filter the rows in the table. Works with Regex too!
Case Sensitive
Key

	Key	Text	Tooltip
1	battlefield_buildings_names_onscreen_name_fro_skv_bridge	Skaven Bridge	✓
2	battlefield_buildings_names_local_effects_description_fro_skv_bridge	The shittiest bridge in all realms, made by retarded sla...	✓

Loc PackedFiles are files that end in ***".loc",*** and contain most of the texts you see ingame. When you open them, you can see they work like a… minimal DB Table. The only real differences with tables is that it doesn't have a "***Clone***" action. Other than that, it's just a little table. Loc PackedFiles uses the same colour code for cells and rows as the TreeView.

Text PackedFiles



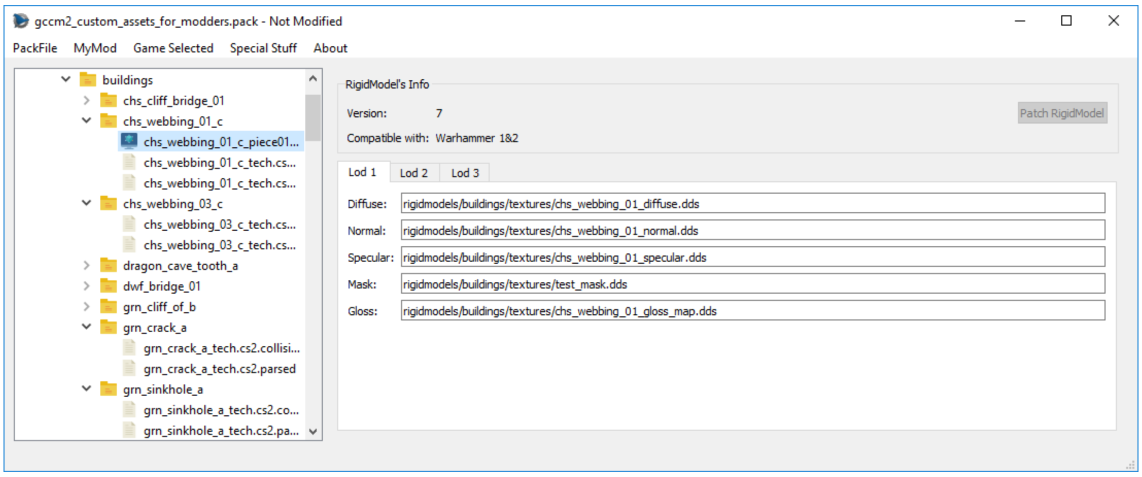
def_hiddenport_def_stalactites_kzd.pack
- terrain
  - battles
    - def_stalactites_kzd
      - icon.tga
      - lf_height_map.data
      - lf_height_map.dds
      - lf_normal.dds
      - lf_sea_height_map.data
      - lf_sea_height_map.dds
      - map_info.xml
      - tile_list.bin
  - tiles
    - battle

```
<map_info>
   <display_name>Dark Elf Hidden Port</display_name>
   <description></description>
   <author></author>
   <team_size_1>1</team_size_1>
   <team_size_2>1</team_size_2>
   <environment>weather/battle/wh_bst_02.environment</environment>
</map_info>
```

RPFM can open and edit a wide variety of Text PackedFiles, such as ***XML, HTML, LUA, TXT,….*** It has native ***Undo/Redo support, Copy/Paste support…*** the normal things for a basic text editor.

Also, exclusive for Warhammer 2 Lua files, there is an option to "***Check Syntax***". This will pass the file through ***Kailua*** (if installed and in the Path) and return you a list of errors encountered. Keep in mind this is experimental, exclusive to Warhammer 2 Lua Files and it may fail.

Rigid Model PackedFiles



- buildings
  - chs_cliff_bridge_01
  - chs_webbing_01_c
    - chs_webbing_01_c_piece01...
    - chs_webbing_01_c_tech.cs...
    - chs_webbing_01_c_tech.cs...
  - chs_webbing_03_c
    - chs_webbing_03_c_tech.cs...
    - chs_webbing_03_c_tech.cs...
  - dragon_cave_tooth_a
  - dwf_bridge_01
  - grn_cliff_of_b
  - grn_crack_a
    - grn_crack_a_tech.cs2.collisi...
    - grn_crack_a_tech.cs2.parsed
  - grn_sinkhole_a
    - grn_sinkhole_a_tech.cs2.co...
    - grn_sinkhole_a_tech.cs2.pa...

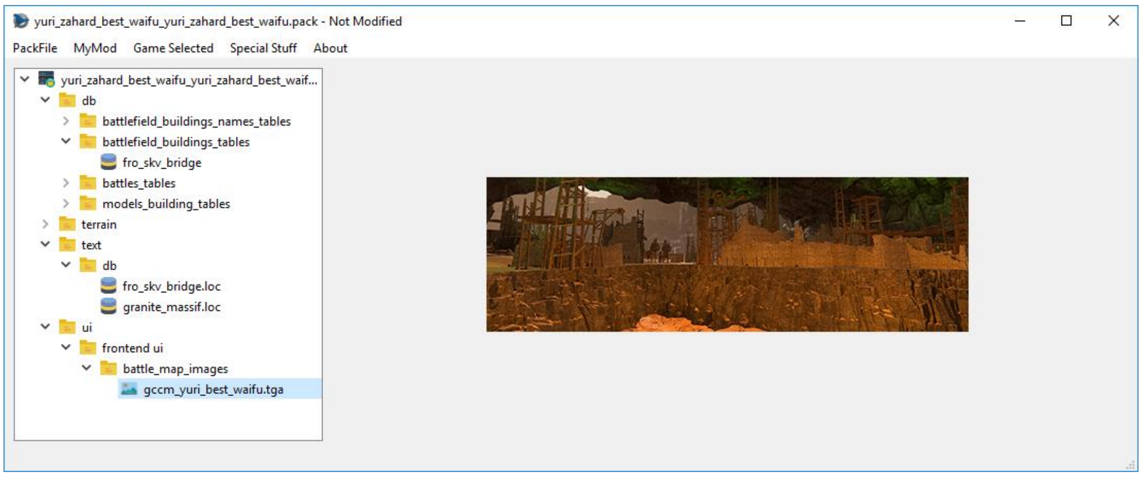
RigidModel's Info

Version:	7
Compatible with:	Warhammer 1&2

Diffuse:	rigidmodels/buildings/textures/chs_webbing_01_diffuse.dds
Normal:	rigidmodels/buildings/textures/chs_webbing_01_normal.dds
Specular:	rigidmodels/buildings/textures/chs_webbing_01_specular.dds
Mask:	rigidmodels/buildings/textures/test_mask.dds
Gloss:	rigidmodels/buildings/textures/chs_webbing_01_gloss_map.dds

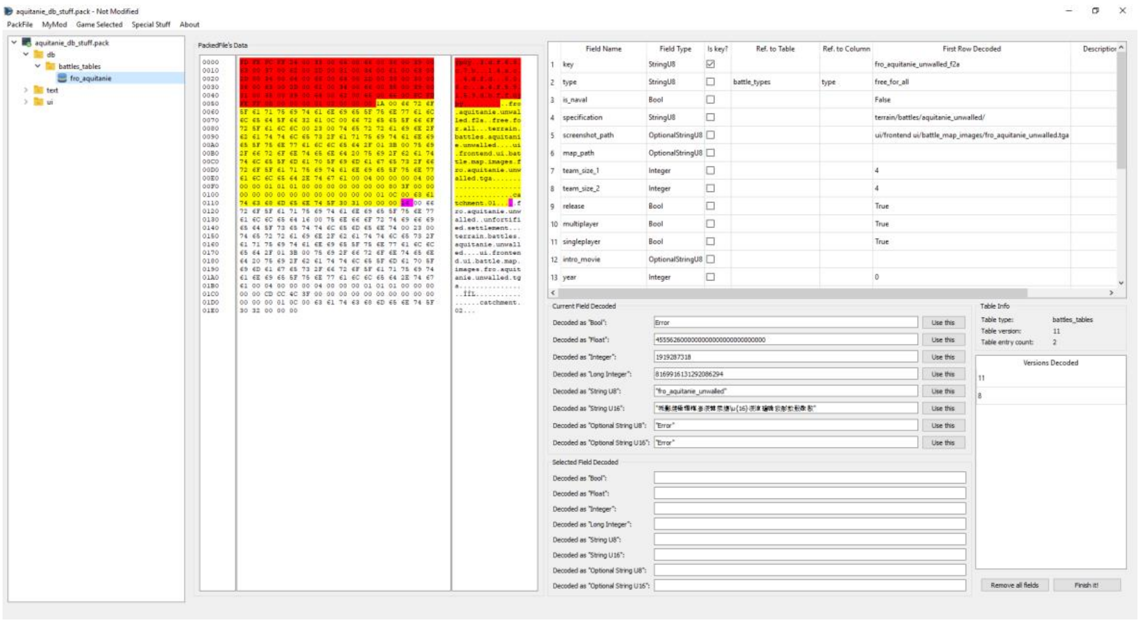
RPFM allows some limited editing of RigidModels (3D models). It allows you to change any of the textures each model uses and to patch Attila's RigidModels for being loadable in Warhammer games (just the model, it doesn't patch collisions or logic). It doesn't work with all the RigidModels.

Image PackedFiles



RPFM can open a variety of image formats, such as ***PNG, JPG, TGA, DDS (most of them)…*** Just select the image you want to see, and it'll open in the right side of the window.

DB Decoder



	Field Name	Field Type	Is key?	Ref. to Table	Ref. to Column	First Row Decoded	Descriptio
1	key	StringU8	☑			fro_aquitanie_unwalled_f2a	
2	type	StringU8	☐	battle_types	type	free_for_all	
3	is_naval	Bool	☐			False	
4	specification	StringU8	☐			terrain/battles/aquitanie_unwalled/	
5	screenshot_path	OptionalStringU8	☐			ui/frontend ui/battle_map_images/fro_aquitanie_unwalled.tga	
6	map_path	OptionalStringU8	☐				
7	team_size_1	Integer	☐			4	
8	team_size_2	Integer	☐			4	
9	release	Bool	☐			True	
10	multiplayer	Bool	☐			True	
11	singleplayer	Bool	☐			True	
12	intro_movie	OptionalStringU8	☐				
13	year	Integer	☐			0	

Current Field Decoded

Decoded as "Bool":	Error	Use this
Decoded as "Float":	45556260000000000000000000000	Use this
Decoded as "Integer":	1919287318	Use this
Decoded as "Long Integer":	8169916131292086294	Use this
Decoded as "String U8":	"fro_aquitanie_unwalled"	Use this
Decoded as "String U16":	[illegible]	Use this
Decoded as "Optional String U8":	"Error"	Use this
Decoded as "Optional String U16":	"Error"	Use this

Table Info

Table type:	battles_tables
Table version:	11
Table entry count:	2

Versions Decoded

11

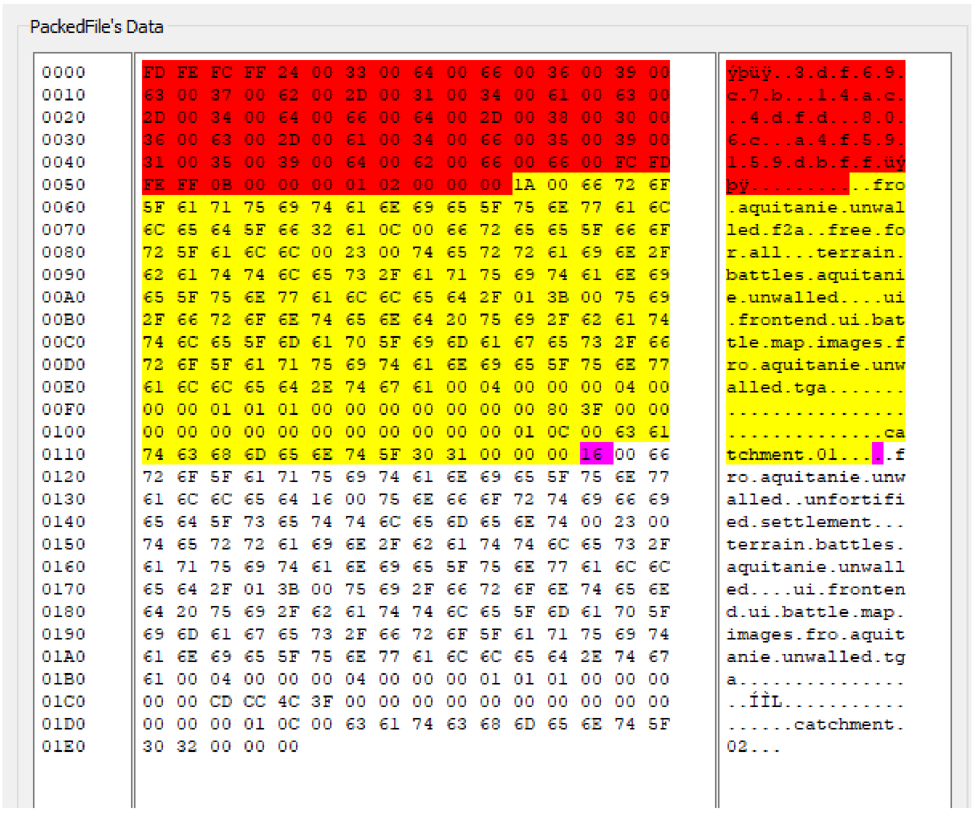
8

Selected Field Decoded

Decoded as "Bool":	
Decoded as "Float":	
Decoded as "Integer":	
Decoded as "Long Integer":	
Decoded as "String U8":	
Decoded as "String U16":	
Decoded as "Optional String U8":	
Decoded as "Optional String U16":	

RPFM has an integrated database decoder, to speed up a lot the decoding process of the "***structure***" (or "***Definition***") of a table. It can be opened by right-clicking on a table and selecting "***Open/Open with Decoder***". Only works on tables.

Starting by the left we have this:


PackedFile's Data

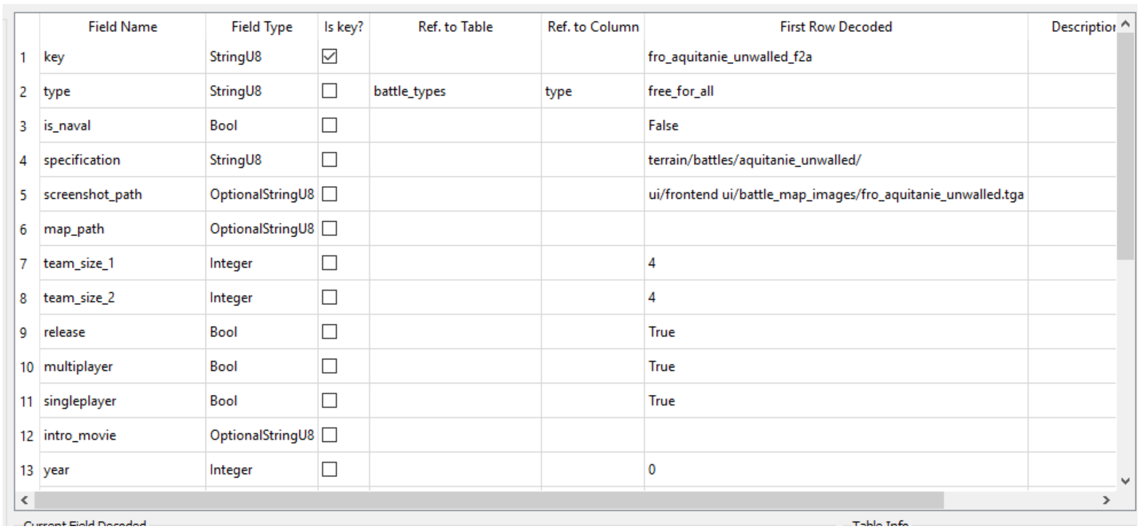
```
0000  FD FE FC FF 24 00 33 00 64 00 66 00 36 00 39 00  ýþüÿ..3.d.f.6.9.
0010  63 00 37 00 62 00 2D 00 31 00 34 00 61 00 63 00  c.7.b...1.4.a.c.
0020  2D 00 34 00 64 00 66 00 64 00 2D 00 38 00 30 00  ..4.d.f.d...8.0.
0030  36 00 63 00 2D 00 61 00 34 00 66 00 35 00 39 00  6.c...a.4.f.5.9.
0040  31 00 35 00 39 00 64 00 62 00 66 00 66 00 FC FD  1.5.9.d.b.f.f.üý
0050  FE FF 0B 00 00 00 01 02 00 00 00 1A 00 66 72 6F  þÿ.........fro
0060  5F 61 71 75 69 74 61 6E 69 65 5F 75 6E 77 61 6C  .aquitanie.unwal
0070  6C 65 64 5F 66 32 61 0C 00 66 72 65 65 5F 66 6F  led.f2a..free.fo
0080  72 5F 61 6C 6C 00 23 00 74 65 72 72 61 69 6E 2F  r.all...terrain.
0090  62 61 74 74 6C 65 73 2F 61 71 75 69 74 61 6E 69  battles.aquitani
00A0  65 5F 75 6E 77 61 6C 6C 65 64 2F 01 3B 00 75 69  e.unwalled....ui
00B0  2F 66 72 6F 6E 74 65 6E 64 20 75 69 2F 62 61 74  .frontend.ui.bat
00C0  74 6C 65 5F 6D 61 70 5F 69 6D 61 67 65 73 2F 66  tle.map.images.f
00D0  72 6F 5F 61 71 75 69 74 61 6E 69 65 5F 75 6E 77  ro.aquitanie.unw
00E0  61 6C 6C 65 64 2E 74 67 61 00 04 00 00 00 04 00  alled.tga.......
00F0  00 00 01 01 01 00 00 00 00 00 00 00 80 3F 00 00  ................
0100  00 00 00 00 00 00 00 00 00 00 00 01 0C 00 63 61  ..............ca
0110  74 63 68 6D 65 6E 74 5F 30 31 00 00 00 16 00 66  tchment.01.....f
0120  72 6F 5F 61 71 75 69 74 61 6E 69 65 5F 75 6E 77  ro.aquitanie.unw
0130  61 6C 6C 65 64 16 00 75 6E 66 6F 72 74 69 66 69  alled..unfortifi
0140  65 64 5F 73 65 74 74 6C 65 6D 65 6E 74 00 23 00  ed.settlement...
0150  74 65 72 72 61 69 6E 2F 62 61 74 74 6C 65 73 2F  terrain.battles.
0160  61 71 75 69 74 61 6E 69 65 5F 75 6E 77 61 6C 6C  aquitanie.unwall
0170  65 64 2F 01 3B 00 75 69 2F 66 72 6F 6E 74 65 6E  ed....ui.fronten
0180  64 20 75 69 2F 62 61 74 74 6C 65 5F 6D 61 70 5F  d.ui.battle.map.
0190  69 6D 61 67 65 73 2F 66 72 6F 5F 61 71 75 69 74  images.fro.aquit
01A0  61 6E 69 65 5F 75 6E 77 61 6C 6C 65 64 2E 74 67  anie.unwalled.tg
01B0  61 00 04 00 00 00 04 00 00 00 01 01 01 00 00 00  a...............
01C0  00 00 CD CC 4C 3F 00 00 00 00 00 00 00 00 00 00  ..ÍÌL...........
01D0  00 00 00 01 0C 00 63 61 74 63 68 6D 65 6E 74 5F  ......catchment.
01E0  30 32 00 00 00                                   02...
```

This is the "***PackedFile's Data***" view. It's similar to a hexadecimal editor, but far less powerful, and it's not editable. In the middle you have "*Raw Hexadecimal Data*", and in the right, you have a "*Decoded*" version of that data. To make it easier to work with it, both scrolling and selection are synchronised between both TextViews. So you can select a byte in the middle view, and it'll get selected in the right one too. The colour code here means:

- ***Red***: header of the table. It contains certain info about what's in the table.
- ***Yellow***: the part of the table already decoded following the structure from the fields table.
- ***Magenta***: the byte where the next field after all the fields from the fields table starts.

For performance reasons, this view is limited to 60 lines, which should be more than enough the decode the first row of almost every table.

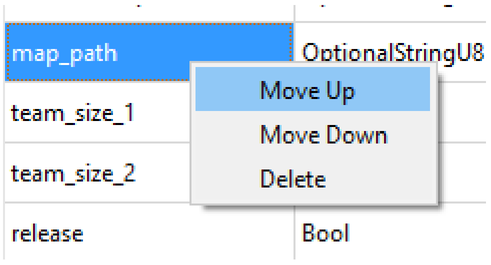
Next, to the right, we have this:



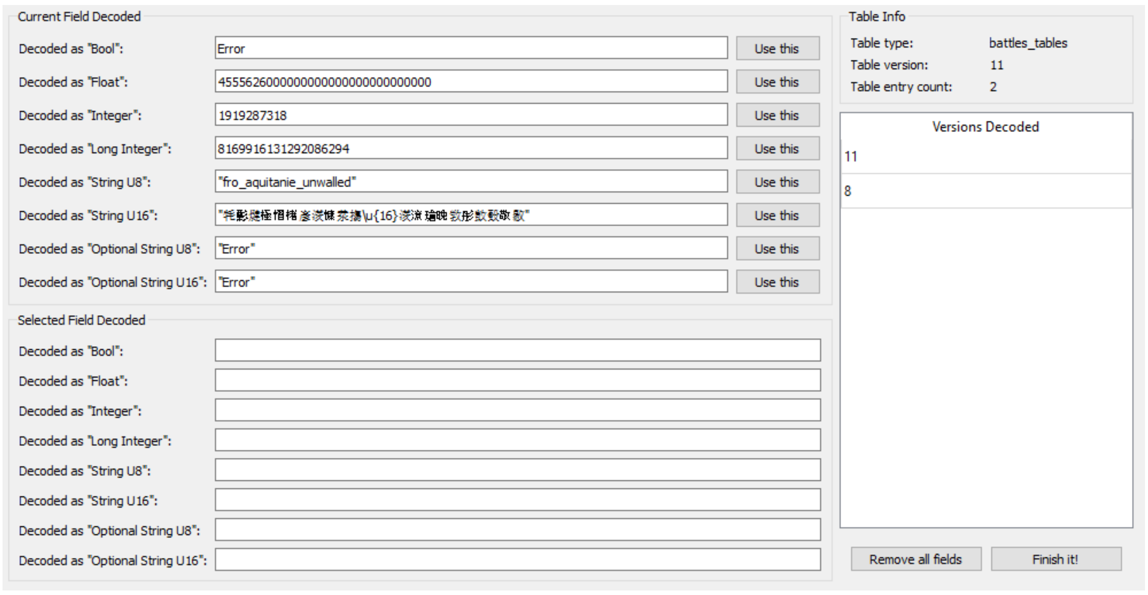
	Field Name	Field Type	Is key?	Ref. to Table	Ref. to Column	First Row Decoded	Descriptio
1	key	StringU8	☑			fro_aquitanie_unwalled_f2a	
2	type	StringU8	☐	battle_types	type	free_for_all	
3	is_naval	Bool	☐			False	
4	specification	StringU8	☐			terrain/battles/aquitanie_unwalled/	
5	screenshot_path	OptionalStringU8	☐			ui/frontend ui/battle_map_images/fro_aquitanie_unwalled.tga	
6	map_path	OptionalStringU8	☐				
7	team_size_1	Integer	☐			4	
8	team_size_2	Integer	☐			4	
9	release	Bool	☐			True	
10	multiplayer	Bool	☐			True	
11	singleplayer	Bool	☐			True	
12	intro_movie	OptionalStringU8	☐				
13	year	Integer	☐			0	

This is the "***Fields List***". Here are all the columns this table has, including their title, type, if they are a "*key*" column, their relation with other tables/columns, the decoded data on each field of the first row of the table, and a "Description" field, to add commentaries that'll show up when hovering a cell of that column with the mouse.

If we right-click in any field of the table, we have these three self-explanatory options to help us with the decoding:



And finally, under the table, we have this:



Current Field Decoded

Decoded as "Bool":	Error	Use this
Decoded as "Float":	4555626000000000000000000000000	Use this
Decoded as "Integer":	1919287318	Use this
Decoded as "Long Integer":	8169916131292086294	Use this
Decoded as "String U8":	"fro_aquitanie_unwalled"	Use this
Decoded as "String U16":	"牦彫臒極楫楮慬湯慷汬敤\u{16}湵慷汬敤 数彫数數敬 敫"	Use this
Decoded as "Optional String U8":	"Error"	Use this
Decoded as "Optional String U16":	"Error"	Use this

Selected Field Decoded

Decoded as "Bool":	
Decoded as "Float":	
Decoded as "Integer":	
Decoded as "Long Integer":	
Decoded as "String U8":	
Decoded as "String U16":	
Decoded as "Optional String U8":	
Decoded as "Optional String U16":	

Table Info

Table type:	battles_tables
Table version:	11
Table entry count:	2

Versions Decoded

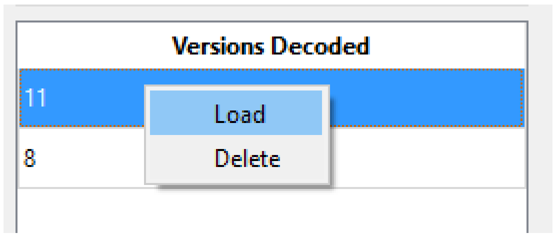
11

8

The "***Current Field Decoded***" will show up the field that starts in the magenta byte of the "*PackedFile's Data*" View, decoded in the different types the tables use. His use is simple: check what type makes more sense (for example, in the screenshot, it's evidently a "*StringU8*"), and click the "***Use this***" button in his row. Doing that will add a field of that type to the "Fields List", and it'll update the "*PackedFile's Data*" View to show where the next field starts. Keep doing that until you think you've decoded the complete first row of the table, hit "***Finish It!***" at the right bottom corner, and select the table again. If the decoding is correct, the table will open. And that's how I ~~met your moth~~er you decode a table.

Under "***Current Field Decoded***" we have "***Selected Field Decoded***". It does the same that "*Current Field Decoded*", but from the byte you selected in the "*PackedFile's Data*" View. Just select a byte and it'll try to decode any possible field starting from it. It's for helping decoding complex tables.

To the right, we have a list of data about the table, and the list of versions of that table we have a definition for. If we right-click in one of them, we can load that version (useful to have something to start when a table gets "*updated*" in a patch) or delete it (in case we make a totally disaster and don't want it to be in the schema).



And at the bottom, we have the "***Remove all fields***" and "*Finish It!*" buttons. The first one clears the "*Fields List*" and reset the decoding process. The second one saves the current "*Fields List*" into the game's schema and reloads the schema, so the changes can be used immediately.