

(ModMaker.chm of rFactor2/support/tools)



gMotor 2.0 Technology

How to make a Mod Package

rFactor 2 (rF2) content is very similar in layout and file types to that used in rFactor, but now must be in the form of mods and components. A component is the collection of files comprising a track, vehicle, HUD, even a UI. All of these files are stored in one or more mas files, and combined into a single deliverable .mod file. A mod is just a collection of components, also combined into a single deliverable .mod file.

Mod files are created using the gMotor 2 MAS File Utility and its two pop-up applications, the Component Packager and the Mod Packager. Both mods and components are installed using one of several different installers: in-game from the UI, using the stand-alone ModMgr utility, or using utilities in both the Component Packager and Mod Packager.

The rF2 retail build uses only packaged components and mods. There are very few loose files anywhere, as almost every file is in a .mas file. Installation of mods consists of extracting the mas files from the component / mod file to specific directories.

rF2 currently supports these components:

- Vehicles

- Locations

- Sounds

- HUD

- Scripts (Pitcrew)

- Commentary

- Talent (.rcd files)

- Helmets

- Showroom
- UIData
- Other (Shared)

Components are located in two directories: Core and Installed. Every component includes a manifest file located in the component sub-directory. Core components are pre-installed into the Core dir. Core components cannot be modified - the game will check these components and won't run if they've been modified. Certain non-core components will also be pre-installed into the Installed dir, along with their manifest files.

Mod files must be located in a dir called Packages under the rF2 root dir, so the in-game mod installer can find them.

A mod consists of at least one Vehicle and one Location component, and optionally others. Installing a mod just means:

- adding its components to the Installed dir, or
- updating a component's manifest files if the component already exists
- adding an rfm mas to the Installed/rFM dir.
- adding a mod manifest file to the Manifests dir

There are three steps to create a Mod after your content is finalized:

1. create and install the components using the Component Packager, or use already installed components. Currently, the Mod Packager tool will only create a Mod from installed components.
2. create an rFm .mas file containing the .rfm file for the mod, and its two associated icon files.
3. combine the components and rfm mas file into a Mod using the Mod Packager.

□

□

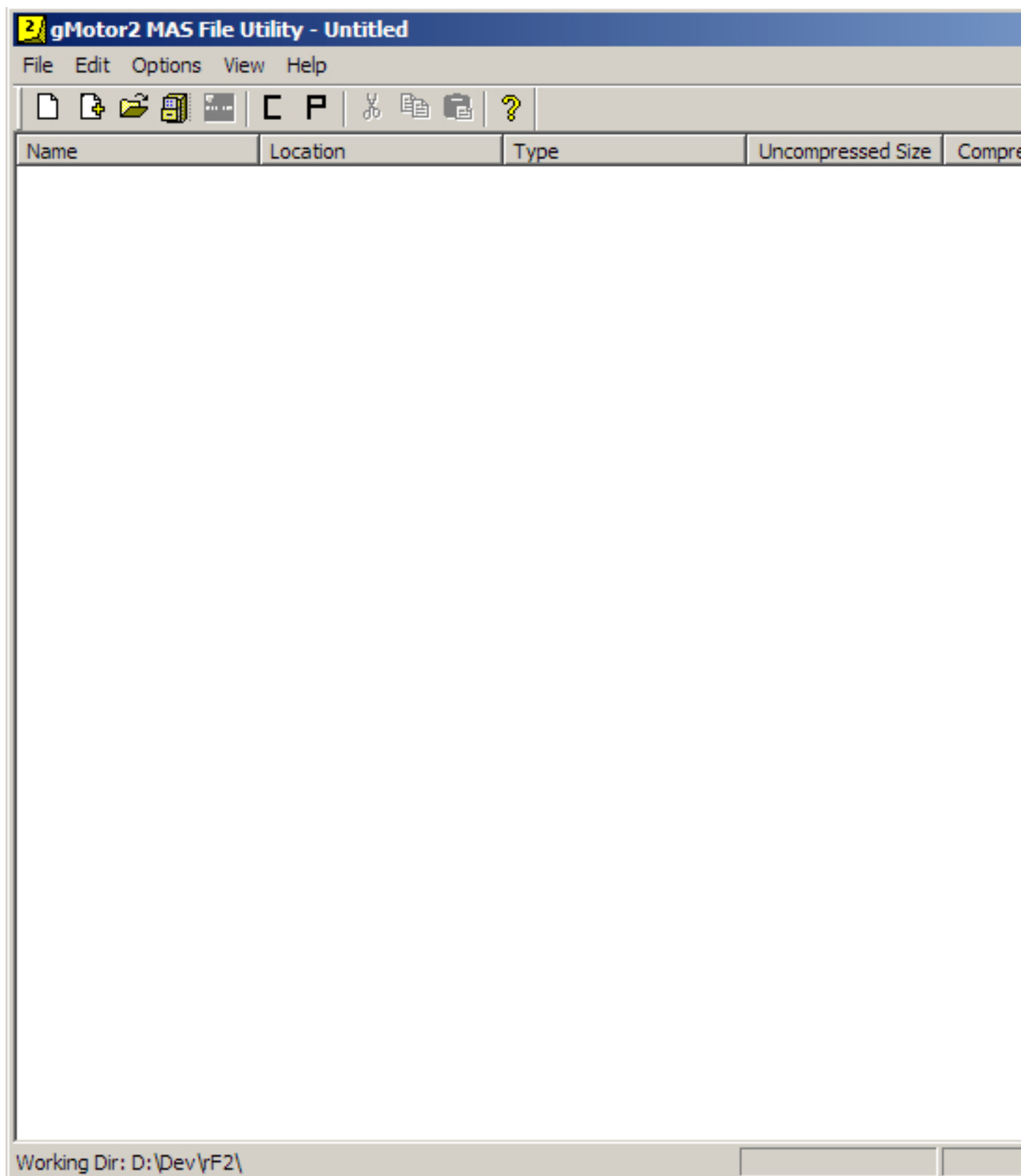
Creating a Component Package

To create a Component, you must first place all of its files into one or more .mas files using the gMotor 2 MAS File Utility. Any file type may be drag-and-dropped into the utility.

Tutorial MAS2.exe ModMaker.chm - How make a mod for rFactor2

Escrito por Avatares Nucleorion

Jueves, 31 de Mayo de 2012 13:26 - Actualizado Jueves, 31 de Mayo de 2012 13:27



You can arrange your mas files any way you want, but vehicle and location components have a couple of special requirements:

- vehicle components must have a **single team-specific mas file** contains all skins and veh files for each team. This mas file should probably be separate, and should be named something unique like c6rteams.mas, not teams.mas, since mas files from a few different vehicle teams could be opened at the same time.

- vehicle components should have (but aren't required to have) a common mas file for all of the non-graphics files, such as the various .ini files, .pm file, .hdv file, .gen files, etc. Try to keep this file small since we will be checking it often, and large files take longer to check.

- location components must have a **separate layout-specific mas for each layout** containing at least the .gdb file for the layout. Each layout-specific mas file should also contain other layout-specific files (.gdb, .aiw, etc.), and each should have a unique name since all layout mas files will be in the same component mas.

An example of a simple and efficient track component would be:

- Racetrack.mas – contains the track spinner icon, .tdf file, or other overall track files.
- Racetrack_GMTs.mas – contains track GMT files.
- Racetrack_Maps.mas – contains track texture files.
- Racetrack_Anims.mas – contains track anim files.
- Racetrack_SponsorArt.mas – contains sponsor-specific track assets
- Racetrack_LayoutA.mas – contains layout-specific art assets, as well as .scn, .aiw, .cam, .gdb, .wet files
- Racetrack_LayoutB.mas - contains layout-specific art assets, as well as .scn, .aiw, .cam, .gdb, .wet files
- Racetrack_LayoutC.mas - contains layout-specific art assets, as well as .scn, .aiw, .cam, .gdb, .wet files

In this case, the _GMTs, _Maps, and _Anims files could be combined into a single mas file

An example of a simple and efficient vehicle component would be:

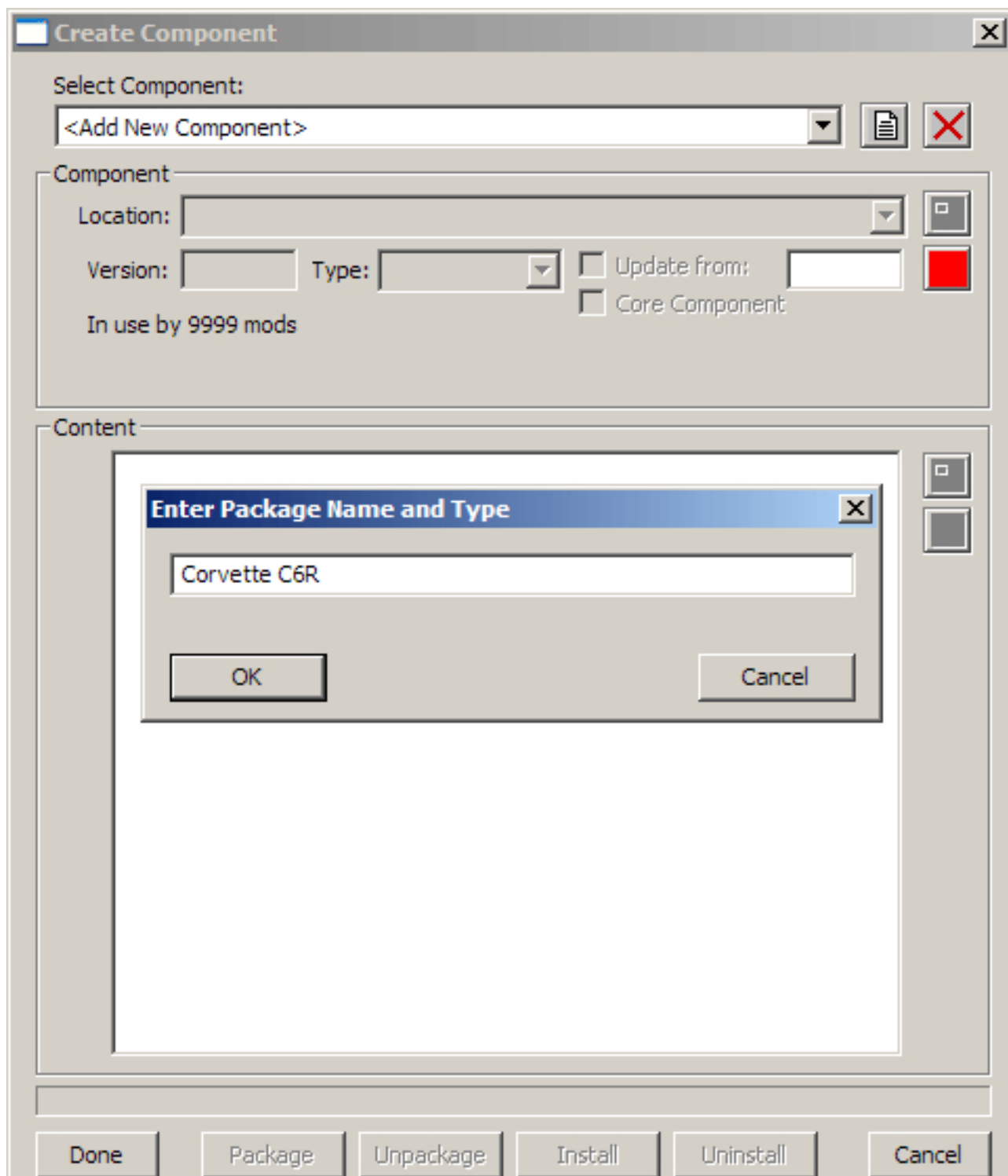
- Racecar.mas – contains GMTs, textures, anims and other geometry or art.
- Racecar_common.mas – contains .ini, .pm, .tgm, .gen, .tbg, etc.
- Racecar_teams.mas – contains team skins, alt skins, and veh files for each team.

Once all component mas files are created, hit the **C** (component) button on the mas utility toolbar to launch the Component Packager. Select and type in a name for the component.

Tutorial MAS2.exe ModMaker.chm - How make a mod for rFactor2

Escrito por Avatares Nucleorion

Jueves, 31 de Mayo de 2012 13:26 - Actualizado Jueves, 31 de Mayo de 2012 13:27

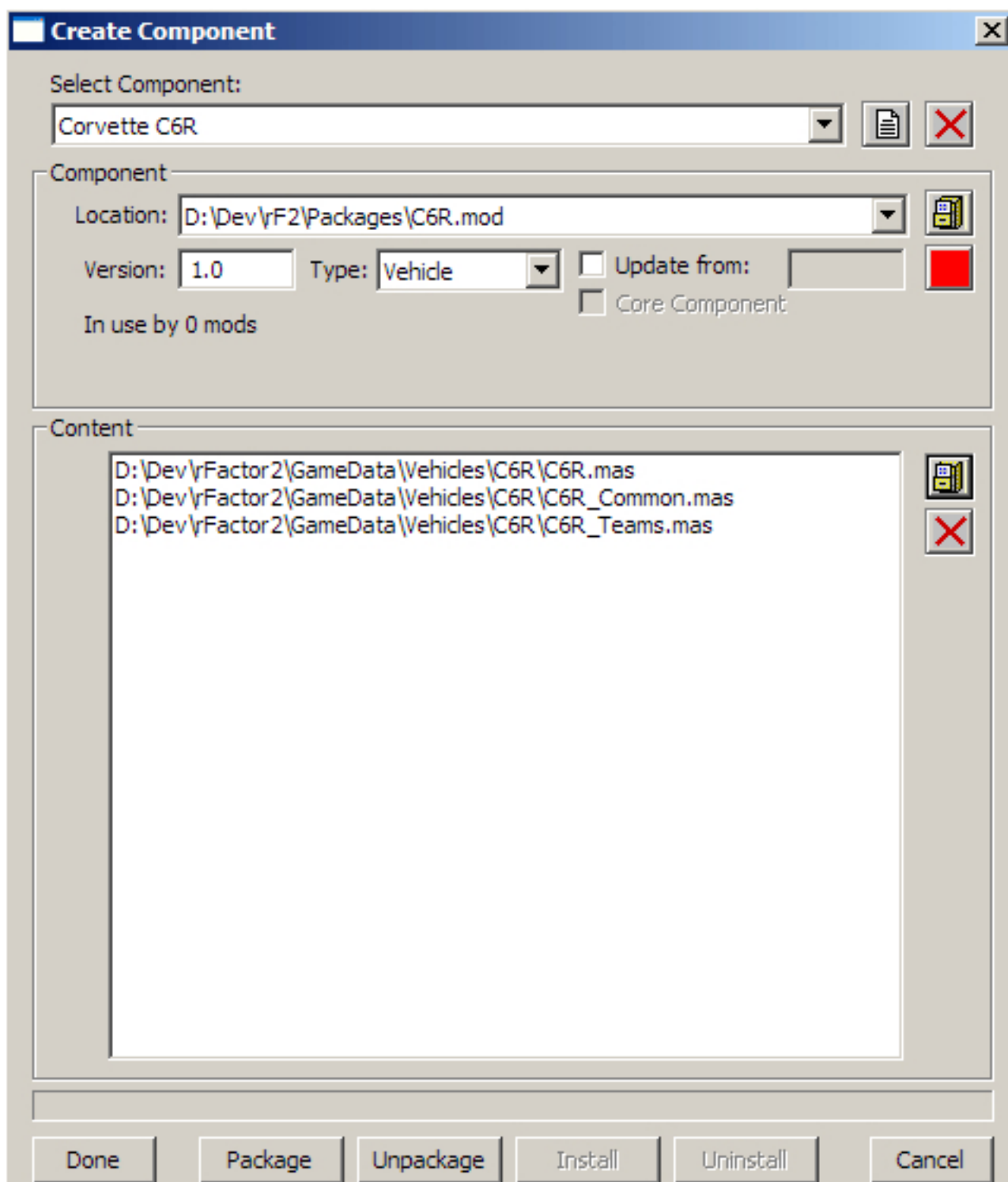


This dialog box is used to create a new component. It prompts you to enter a package name and type, and then to select a file to use as the component's data file. The 'Package' button is used to create a new package, and the 'Unpackage' button is used to remove a package. The 'Install' button is used to install a component, and the 'Uninstall' button is used to remove a component. The 'Cancel' button is used to cancel the operation.

Tutorial MAS2.exe ModMaker.chm - How make a mod for rFactor2

Escrito por Avatares Nucleorion

Jueves, 31 de Mayo de 2012 13:26 - Actualizado Jueves, 31 de Mayo de 2012 13:27



When you're finished adding mas files, hit the **Package** button. The installed component will be added to the

Working Dir: D:\Dev\F2\
When you're finished adding mas files, hit the **Package** button. The installed component will be added to the

You can also install components using the ModMgr utility, or directly from the Mod Packager popup.

Creating an rFM MAS File

Using the gMotor 2 MAS Utility, create an rFM mas file containing the rFM file for the mod, and the two icon files *Icon.ext and *SMIcon.ext, where .ext represents the extension for an image file in .bmp, .tga, or .dds format.

Creating a Mod Package

Creating a Mod package requires an initial step of creating an rfm mas file, which contains the rfm file for the mod, and the two icon files for the UI mod spinner. The rfm file is really just a placeholder - most of the info in the .rfm file will still be used except for the track and car filters. You will manually add track layouts and vehicle teams to the mod in a later step. You can save the rfm mas file anywhere.

Once the rfm mas file is created, hit the **P** (package) button on the mas utility toolbar. All you can do at this point is hit the Select rFM Browse button to locate your rfm mas file. The Mod Name is taken from the rfm file in the rfm mas, and it's probably best not to change this. If you want a different Mod Name, change it in the original rfm file, re-mas, and select the new version. Select the mod file location, and enter a version string.

Tutorial MAS2.exe ModMaker.chm - How make a mod for rFactor2

Escrito por Avatares Nucleorion

Jueves, 31 de Mayo de 2012 13:26 - Actualizado Jueves, 31 de Mayo de 2012 13:27

Create Package

Select rFM: D:\Dev\rfactor2\rfm\C6R.mas

Mod Name: GT World Endurance Championship

Package

Location: D:\Dev\rf2\Packages\C6R.mod

Version: 1.0 ☐ Update from:

Content

Locations	Version
-----------	---------

Layouts

Vehicles	Version
----------	---------

Teams

Other	Version	Type	Installed
-------	---------	------	-----------

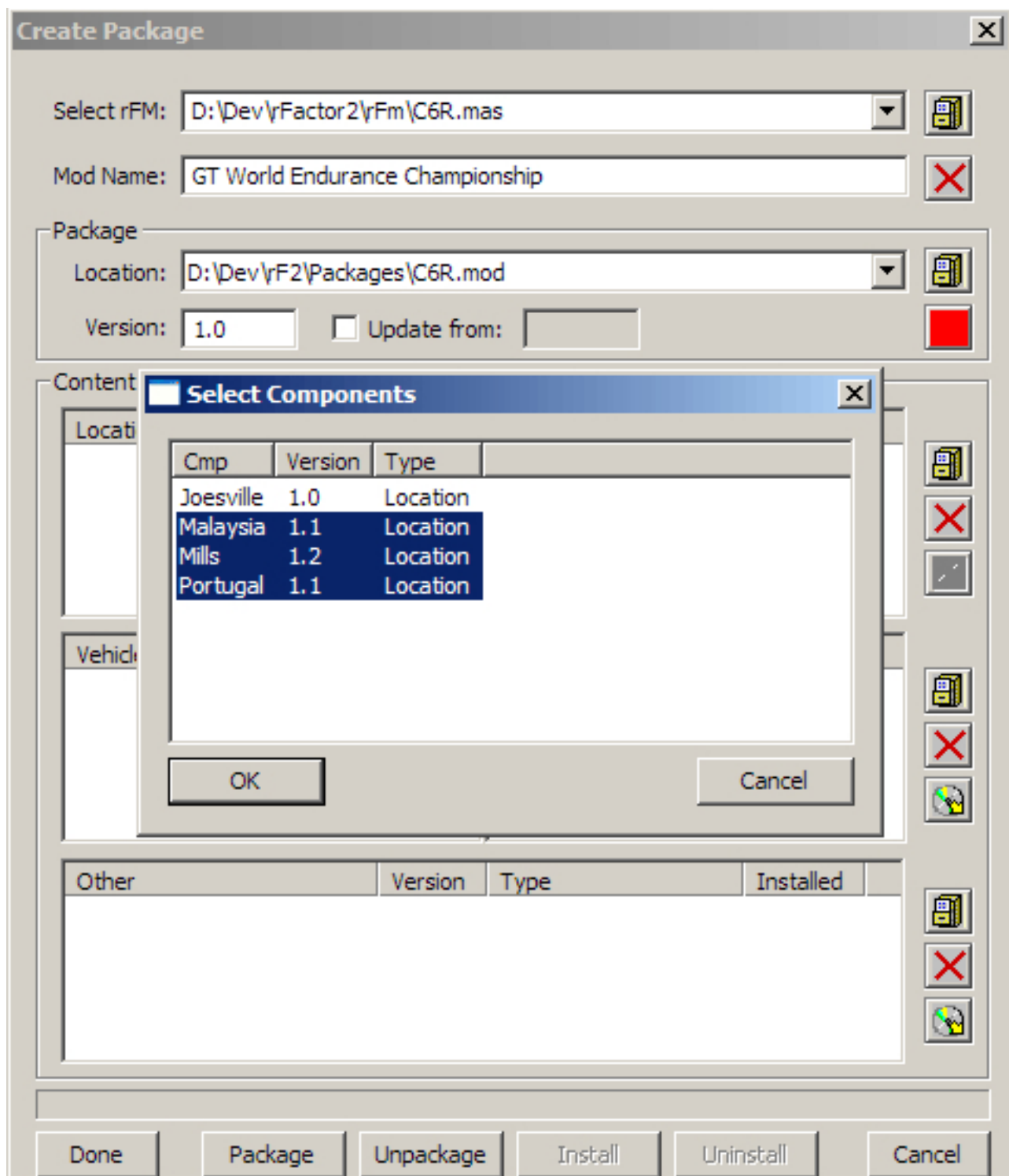
Done Package Unpackage Install Uninstall Cancel

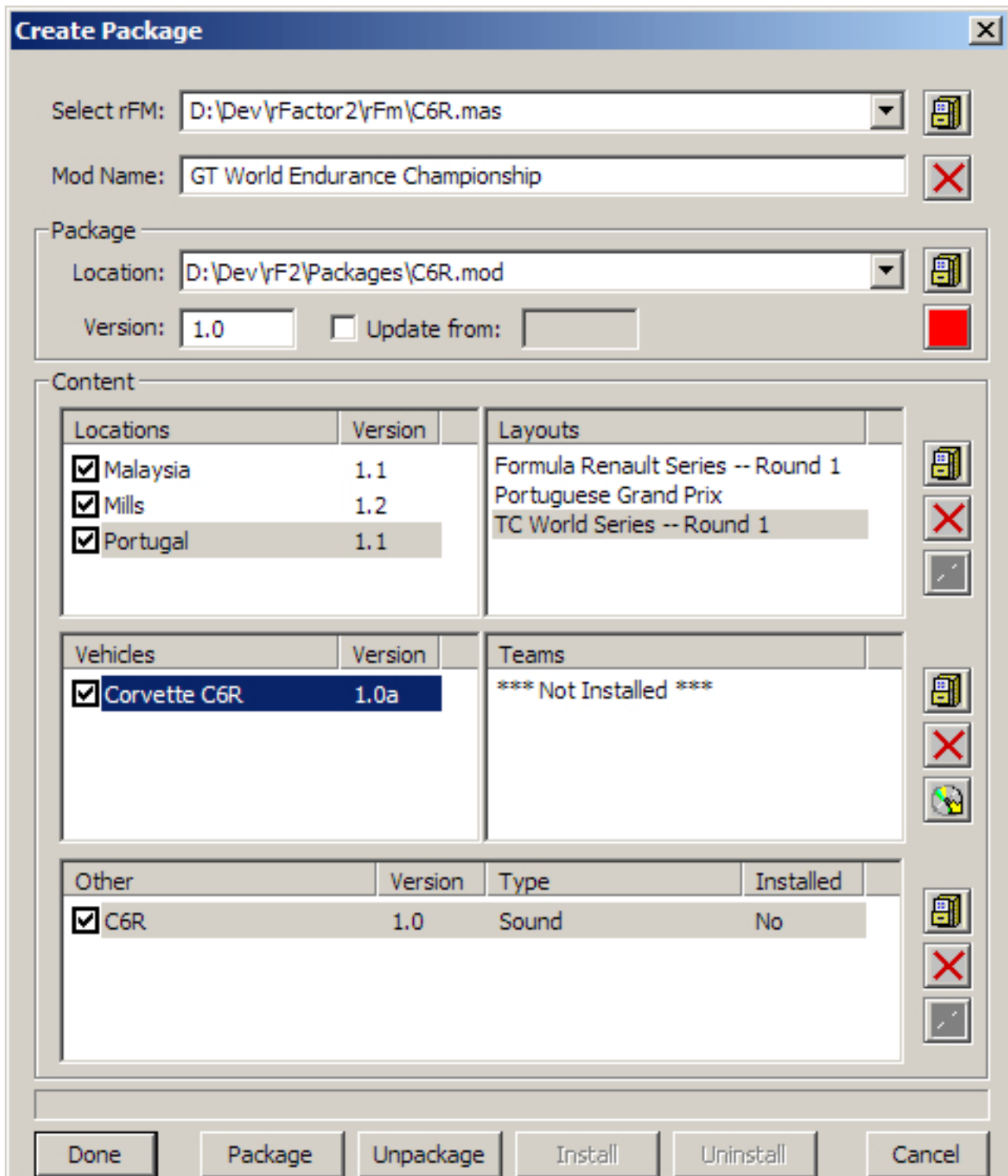
Now select the locations, browse the components list of possible locations/vehicles/teams. The

Tutorial MAS2.exe ModMaker.chm - How make a mod for rFactor2

Escrito por Avatares Nucleorion

Jueves, 31 de Mayo de 2012 13:26 - Actualizado Jueves, 31 de Mayo de 2012 13:27





Now just hit Package, and once the process completes, you will have an installable mod file.

Advanced Features

Updates

Mod packages are usually stand-alone, which means that all files required for the mod are contained in that .mod file, and in the Installed subdirectory for that mod. It is also possible to create an Update when you only want to change a few files, and don't want to redistribute an entire mod package. An Update is a mod/component file containing only mas files with updated content, and is dependent on an existing mod. You create an update by checking the Update From: box, and entering the version of the existing mod required for the update. The mod version must be different from the dependent version. Updates will install alongside the existing mod, and will use any mas files from the existing mod that are not part of the update.

Since all files in a component are in mas files, it is not possible to have an Update with just a single file, you must include the entire mas file containing the updated file and its remaining original files.

Virtual Components

Virtual components are another way to distribute a mod with a minimal file size. By unchecking the box next to a component, you are indicating that this component is part of your mod, but the actual files are not part of your distribution. When a user installs a mod with virtual components, the exact component must already be installed on their machine, or the install will fail.

Tutorial MAS2.exe ModMaker.chm - How make a mod for rFactor2

Escrito por Avatares Nucleorion

Jueves, 31 de Mayo de 2012 13:26 - Actualizado Jueves, 31 de Mayo de 2012 13:27
