

Table of Contents

Updating a Module	1
-------------------------	---

Updating a Module

Inevitably, your module will need to be updated. A new feature or functionality could be added to the VASSAL engine; you may get a better idea on how to implement one of your module's features; or the game itself may be produced in a new edition.

To update a module,

1. In the Module Manager, right-click the module and pick **Edit Module**.
2. In the Configuration Window, right-click the **[Module]** node and pick **Properties**.
3. In **Version Number**, enter a version number higher than the existing one. Click **Ok**.
4. In the Module Editor, make your edits as needed.
5. Click **File** | **Save As**. Save the updated module with a new filename to reflect the new version number.

Update Guidelines

Follow these guidelines when updating your module.

- **Latest Version:** Make sure you have the latest version of the VASSAL engine installed when updating your module.
- **Version Numbering:** The version number of a module update must be higher than the one used in an earlier version. Some designers like to use the major/minor version numbers (x.y) common in the programming world. A change to the major version number reflects large changes to the module; a change to minor version reflects smaller ones. For example, a change from 1.0 to 2.0 reflects a major revision (perhaps a new board or toolbar), while 1.0 to 1.1 reflects a lesser update (such as fixing a typo). You can revise the version number in the **Properties** of the **[Module]** node.
- **Filename:** Always include the new module version number in the updated filename, so players can quickly tell which version they have without opening it. For example, chess_2.6.vmod would indicate version 2.6 of a chess module. (Use the **Save As** button to save the module with a new filename.)
- **Graphics:** When updating graphics, if possible, use the same filenames as graphics that exist in the module already. This way, the new graphics will *replace* the old ones in the module file, rather than adding new, unnecessary files, which can cause the module to become bloated.
- **Saved Games:** Remember to update any saved games (Pre-Defined Setups) that the module includes. See *I'm Not Seeing My Changes*, below, for more explanation. Remember to add the updated saved games to the module.
- **Update Extensions First!** Making major changes to the structure of a base module can cause issues with editing any extensions associated with the module, particularly when the extension

depends on components from the base module.

- o. For example, you revise a Game Piece Palette in the base module from a Tabbed Panel to Scrolling List. However, there are pieces in an extension that are assigned to the Tabbed Panel, which no longer exists. In this case, the extension would fail to open because it would depend on the structure of the base module, which has changed.
- o. If you plan to make revisions to a module that includes extensions, always edit the extensions first before editing the base module. In the example above, you might create a temporary palette in the Extension that would correspond to the new structure of the base module, and assign the pieces to it. Then you could edit the base module with the new palette. Finally, you could re-edit the extension and assign these pieces to the new palette.

I'm Not Seeing My Changes

If you don't see your edits reflected in the game, this may be because you are actually looking at a saved game, not the actual module. If your module includes saved games (Pre-Defined Setups), then after making updates, you will not see those

Updating a Module: Extensions and Changing Filenames

changes take place in the module until you update the saved games. (This is a common oversight among module designers when updating modules. See *Pre-Defined Setups* on page 97 for more information.)

To rectify this, launch your module and load the saved game. Make your updates to the saved game, adding any newly defined pieces or other components, and save it again. Then re-add the updated saved games to the module. (You may need to use the Saved Game Updater to make sure that any Game Piece updates are reflected in the new saved game.)

Extensions and Changing Filenames

Extensions for a module are located in the directory <Module File Name>_ext. For example, extensions for version 1.0 of the *Global Thermonuclear War* module, named gtw_1.0.vmod, are located in the directory gtw_1.0_ext.

If you change the file name when updating and include the new version number in the filename (such as to

gtw__2.0.vmod), then Module Manager will no longer be able to locate the extension files from the old version of the module.

As a result, the extensions will not load automatically with the new version. This is easy to fix, but players will need to repair

this individually in their own Module Manager.

To re-add existing extensions to the updated module,

1. In Module Manager, locate the new version of the module. (Check the **Module Version** column for version number.)

2. Right-click the new version and pick **Add Extension**.
3. Browse to, and then select, an existing extension file.
4. Repeat Steps 2-3 for any other extensions.

VASSAL will re-locate the existing extension files from the old directory <OldModuleName>_ext to a new directory named <NewModuleName>_ext. (The new extension will be created on the player's system automatically.)

Each extension will be activated by default. For more information on activating and deactivating extensions, see the *VASSAL User's Guide*.

New Versions of Existing Modules

As an open-source project, VASSAL modules are freely editable by others. If you've got an idea to improve an existing module, either for your own use or that of others, have at it! You may want to contact the module designer and propose a collaborative effort, or you may wish to use the existing module as a starting point for your own new version. There can be any number of VASSAL modules for a given game.

It's always courteous to inform the original module designer or maintainer of your efforts. Contact email addresses of module contributors are maintained on each module's page.