

Table of Contents

Creating Module Extensions	1
----------------------------------	---

Creating Module Extensions

An *extension* is a file that adds features or components to a module. For example, for the fictional game “World War II”, the basic module might include 6 maps, each one the scene of an important European battle. Later, the “North Africa” extension adds 2 more maps that show the scenes of two important North African desert battles, as well as counters representing new tank models. When the extension is loaded, players can select from the older maps or the newer maps, and use the new counters as well.

A module can include any number of extensions.

You should be familiar with creating and editing modules before attempting to create or edit an extension.

What an Extension Can Do

An extension can add functionality or components to a module. Here are some examples of what an extension could add to a game:

- New Game Pieces (units, cards, or any other pieces).
- New Cards to a Deck.
- New maps or boards.
- New tools, such as die rollers or charts.
- New Prototype Definitions (see Extension Prototypes in the Base Module, below).

What an Extension Can't Do

An extension cannot do any of the following:

- Add new Traits to Game Pieces already defined in the base module.
- Remove, disable, or modify existing components from the base module (such as boards, buttons, and other controls).
- Modify, replace, or override Prototype Definitions from the base module.
- Add new Sides.

Using the Extension Editor

An extension is created using a special editor called the Extension Editor, which is similar to the Module Editor.

In the Extension Editor, the process of adding components is the same for extensions as it is for modules: right-click on a node of the tree, select the component you wish to add, and specify its

settings. Note that in the Extension Editor, the components of the main module are displayed, but are disabled, indicating that they may not be modified or deleted by the Extension Editor.

Extensions, and components defined in them can refer to global items defined in the base module, such as Prototypes, Global Properties, Global Key Commands, and Global Hotkeys. For example, if the base module included a Prototype definition called *Zombie*, pieces created in the extension could be assigned the *Zombie* Prototype. (This is a two-way street. See *Extension Prototypes in the Base Module* on page 107 for more information.)

Extensions are given the extension `.vmdx`. Like modules, they are ZIP files and can be decompressed using any ZIP-capable utility.

Extension Properties

Each extension has the following Properties.

- **Version Number:** Revision number of the extension.
- **Description:** Brief description or title of the extension. This will be displayed in the Module Manager.

Creating Module Extensions: Using the Extension Editor

- **Extension ID:** An extension ID links counters in existing save games to the counter definitions in the extension. If this ID is changed, then the Saved Game Updater may not be able to update the counters from existing save games.

Universal Extensions

Although extensions are usually intended to enhance a specific module, it is possible to create a universal extension that can be used by any module.

For example, you could create a universal extension named `Percentile_Dice` includes a die roller which randomly generates a number from 1-100. You could then use this extension for any module that requires random numbers be generated in such a range.

To create an extension,

1. In Module Manager, right-click on the module for which you wish to create an extension. Then, select **New Extension**. The Extension Editor opens.
2. On the Extension Editor Toolbar, click **Extension Properties**.
3. In the **Extension Properties** dialog, do the following:
 - a. Enter a version number and description of the extension.
 - b. ☐ If desired, enter an Extension ID.
 - c. ☐ If you wish this extension to be used with any module, select **Allow loading with any module**.
 - d. Click **Save** to save the Properties.
4. In the Extension Editor, add or edit components as desired.

5. On the Extension Editor Toolbar, click **Save**. Save your extension with the suffix .vmdx.

To edit an extension,

1. In Module Manager, select the extension you wish to edit. Right-click and pick **Edit Extension**. The Extension Editor opens.
2. Add or edit components to the extension as desired.
3. On the Extension Editor Toolbar, click **Save**. Save your extension with the suffix .vmdx.

Copying, Pasting and Editing Module Components

One useful technique for adding new components to an extension is to copy and paste a similar component from the base module into the corresponding node of the Extension Editor, and then edit the copy as required. Although you cannot change the base module, you can utilize its components in the Extension Editor. (The same strictures apply for pasting as they do for the Module Editor; you can only paste like to like. For example, you could paste Game Pieces from the module palette into an extension palette, but you could not paste a Game Piece copied from the module to a turn counter in the extension.)

Integrating Extensions

Although an extension cannot itself modify the components in the base module, it's sometimes useful to manually modify the base module, in the Module Editor, to make room for extension functionality and improve integration.

Extension Toolbar Menu Items

An extension may add new buttons to the base module's Toolbar. Ordinarily, these new buttons would be displayed separately from the module Toolbar buttons. However, if want to add them to an existing module Toolbar menu, simply add the names of these buttons to the Toolbar Menu component of the main module, in the Module Editor.

For example, the main module includes a Toolbar Menu named Maps that includes the game's 3 basic maps: Map 1, Map 2 and Map 3. You later create an extension with Map 4. In the base module, in the Module Editor, you would add *Map 4* to the Toolbar Menu component. Now the Toolbar Menu would include all of the game's maps (1 through 4). However, the Map 4 menu item would not be displayed until the Map 4 extension is loaded.

Creating Module Extensions: Using the Extension Editor

Extension Prototypes in the Base Module

Prototypes in the base module are usable by pieces in the extension, and Prototypes defined in an extension will be available to pieces in the base module when the extension is loaded. This enables you to add optional functionality to the base module, which would be activated by loading an extension, and requires that you edit the base module.

For example, we add a Prototype trait called Extension1 to every counter in a base module. However, no Prototype named Extension1 is defined in the base module. When a user uses the base module with no extensions loaded, the Prototype Trait Extension1 is ignored because the definition

does not exist in the module, and it has no effect on game play.

We then create a Prototype Definition named `Extension1` in an extension with the appropriate Traits. When the base module is used with the extension, all counters defined in the base module will now have the extended Traits defined in the `Extension1` Prototype.

Testing Your Extension

If the Extension Editor is open, launching new games will launch the base module with the extension loaded (automatically activated), enabling you to test the extension like you would a module.

You can only test one extension at a time this way. To test multiple extensions together, you will need to close the Module Editor and the Extension Editor, and launch the game from the Module Manager into regular play mode.

Activating an Extension

In order for a player use an extension, it must be *activated*. For information on activating extensions, consult the *VASSAL User's Guide*.

Example: Creating an Extension for a Card Game

Card-based games often include expansion sets that increase the number and variety of cards available for play. Creating an extension for such expansion sets is straightforward, particularly if the extension requires no new rules or game functionality.

You should be familiar with working with the Extension Editor, before attempting to create an extension for a card game.

Scan, create, or otherwise acquire all of the graphic images for your new cards before beginning.

1. Open the base module in the Extension Editor.
2. In the Extension Editor, locate the card deck (**[Deck]** node) you wish to add cards to. (It will appear disabled and grayed-out).
3. Expand the view of the **[Deck]** node to display the cards in the deck.
4. Right-click a sample card in the deck and pick **Copy**.
5. Right-click the **[Deck]** node and pick **Paste**. You will now be able to edit the pasted card to reflect a card from the expansion. You can change the card name or basic image, add new Traits or Prototypes, or otherwise edit the new card as needed.
6. Repeat Steps 4-5 for any remaining new Cards from the expansion.
7. On the Extension Editor Toolbar, click **Save**. Save your extension with the suffix `.vmdx`.

You can now test and activate your extension.

To add complexity or new functionality, your extension could include new Prototypes to reflect new types of cards available in the extension.