

Developing Extensions for BASINS 3.0

BASINS 3.0 features a completely reengineered system architecture. Unlike its predecessor, all functional components of BASINS 3.0, such as model interfaces, data management utilities, and watershed assessment tools, are now ***BASINS extensions***. The ***BASINS Extension Manager*** loads and unloads extensions from a directory structure maintained separate from ArcView extensions. This architecture allows users to load only extensions needed for their BASINS project, reducing both project size and loading times. The new architecture also allows the system to support several levels of hardware and software sophistication (e.g. users only need to acquire the ArcView Spatial Analyst extension if they intend to use BASINS components requiring that extension). In addition, the extension architecture is easier to maintain and upgrade in that updates for individual extensions rather than the entire system can be issued.

The purpose of this document is to instruct third party developers in how to create extensions compatible with BASINS 3.0. In order for these extensions to work with the ***BASINS Extension Manager***, developers must do two things:

1. Use special calls for adding and removing menus within the extension install and uninstall scripts; and
2. Place the extension in the appropriate folder in the BASINS directory structure.

These two procedures are described below.

Adding and Removing BASINS Menus and Menu Choices

A special procedure should be used when adding menus and choices to BASINS. First, the menu choice(s) for an extension must either go into one of the existing BASINS menus in the View GUI, or be placed in a new folder, with the same name as the desired menu name. The available BASINS menus are:

- Assess
- Data
- Delineate
- Models
- Reports
- Utilities

The ***install script*** for a BASINS extension uses a special command for adding menus and menu choices. In place of the standard ArcView request to add a menu or menu choice [(aMenuBar.Add(aMenu) or aMenu.Add(aMenuChoice)], you should use the following line of code:

```
av.run("BASINS3.MenuAdd",{ aMenu, aMenuChoice })
```

This line calls the `BASINS3.MenuAdd` script. The `BASINS3.MenuAdd` script checks to see if the menu already exists. If not, it creates the menu. If the menu already exists, the script just adds the menu choice to the existing menu. The script also maintains the various `BASINS` menus in the proper order, regardless of the order in which `BASINS` extensions are turned on.

The `BASINS3.MenuAdd` script receives two parameters, (1) the Menu object and (2) the menu choice (menu item) object. Here is an example of an *install script* using the `BASINS3.MenuAdd` code:

Install Script:

```
theMenuObject = SELF.Get(0)
theChoiceObject = SELF.Get(1)

av.run("BASINS3.MenuAdd",{theMenuObject,theChoiceObject})
```

Similarly, the *uninstall script* for a `BASINS` extension uses a special command for removing menus and menu choices. In place of the standard ArcView request to remove a menu or menu choice [(aMenuBar. Remove(aMenu) or aMenu.Remove(aMenuChoice)], you should use the following line of code:

```
av.run("BASINS3.MenuRemove",{ aMenu, aMenuChoice })
```

This line calls the `BASINS3.MenuRemove` script. This script checks if a menu choice is the last remaining choice in a `BASINS` menu. If so, it will remove the menu as well as the choice.

The `BASINS3.MenuRemove` script receives the same two parameters as the `BASINS3.MenuAdd` script, (1) the Menu object and (2) the menu choice (menu item) object. Here is an example of an *uninstall script* using the `BASINS3.MenuRemove` code:

Uninstall Script:

```
theMenuObject = av.GetProject.FindGUI("View").GetMenuBar.FindbyLabel("Data")
theChoiceObject = theMenuObject.FindbyLabel("GenScn")

av.run("BASINS3.MenuRemove", {theMenuObject, theChoiceObject})
```

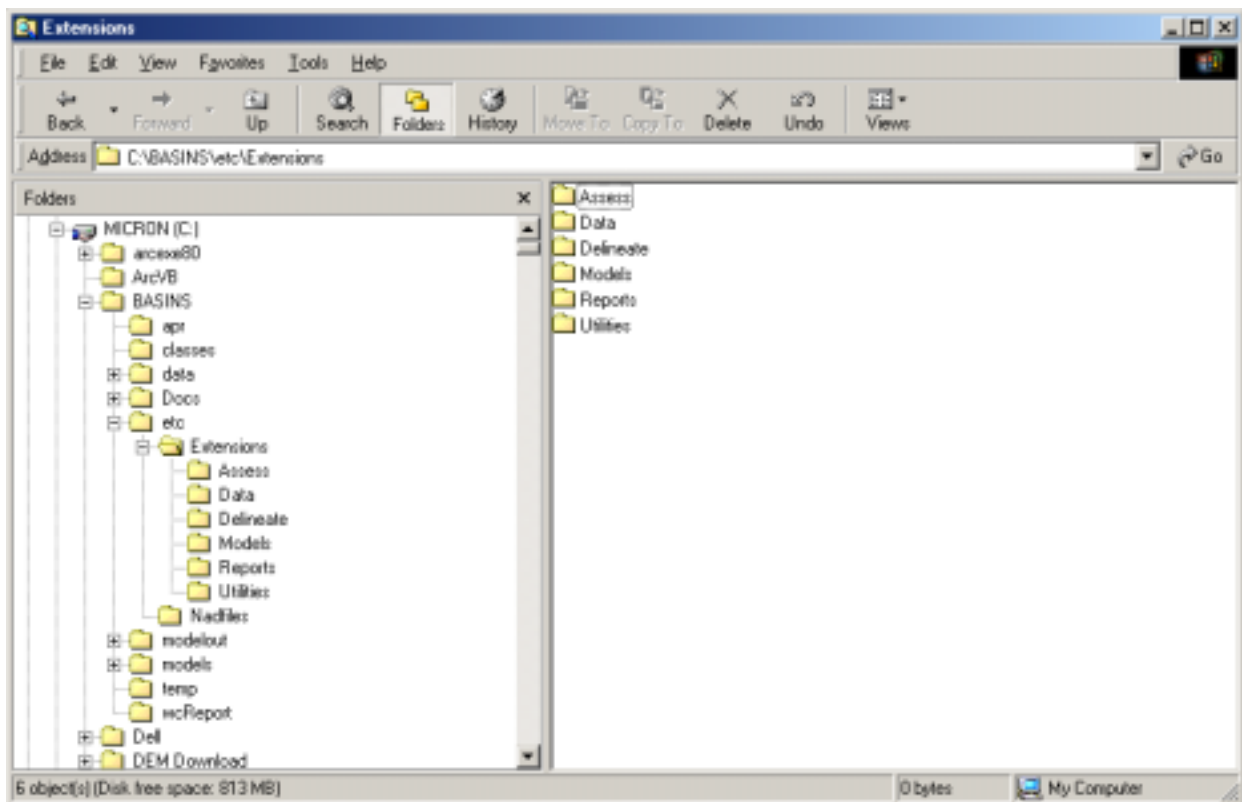
There are no special commands for adding buttons, tools, or other GUI components. These should be handled the normal way for ArcView extensions.

Directory Structure for BASINS Extensions

BASINS 3.0 extensions are stored within the BASINS system directory structure for access by the *BASINS Extension Manager*. Storing BASINS extensions within this structure allows for a cleaner installation and removal of BASINS applications. The *BASINS Extension Manager* organizes BASINS extensions into categories (Assess, Data, Delineate, Models, Reports, and Utilities), and positions BASINS menus and menu choices correctly within the ArcView GUI.

BASINS extensions are stored in the path //BASINS/etc/Extensions/. Within the Extensions folder there is a subfolder for each extension category as shown in Figure 1.

Figure 1. Directory Structure for BASINS Extensions



The subfolders under the Extensions folder correspond to the BASINS Menus - Assess, Data, Delineate, Models, Reports, and Utilities. Developers should determine under which menu category their extension best fits, or create a new menu folder, and place their extension in the associated folder. Any other files used by the extension (e.g. DLLs, data files, etc.) should be placed within the same folder. These associated files can easily be located by getting the extension file's file path using the code:

```
Extension.GetFileName.ReturnDir
```

If the procedures described above are followed, the new BASINS extension will appear in the *BASINS Extension Manager* and will install and uninstall correctly within BASINS.