CHAPTER 6

Customizing NNM from the GUI

One compelling aspect of NNM is that the administrator can customize submaps from within the GUI. Some of the customizations include adding background graphics, saving window placement, hiding symbols, and creating MIB applications. While advanced customizations require file editing, NNM provides the flexibility to offer both simple and advanced customizations without source code modification and re-compilation. In fact, many modifications can be made on the fly without exiting the NNM GUI or restarting processes.

Sealey Inc. has a requirement to be able to make use of geographical bitmaps and customer floor plans. They would like to place certain devices in a specific region representing the physical locations of the devices. Certain devices are mission critical and they would like to configure the GUI with quick access to these devices. However, because each operator has different responsibilities, they consider different devices to be mission critical. Therefore, each operator may want a slightly different view of the network.

Sealey Inc. would also like to be able to graph SNMP MIB data real-time. Certain operators may frequently be required to access data from the same MIB variables. The network application should provide the ability to easily graph MIB data. This chapter details customizing NNM from the GUI.

6.1 CUSTOMIZING SUBMAPS

Submaps may be customized on a per-map basis to allow each NNM user to focus on specific needs. For example, a particular user might want to see the systems placed geographically on a submap according to the location in which the systems physically reside. A user might also want to create custom icons that will launch frequently used applications. The following sections include some of the commonly used map customizations.

6.1.1 Setting the Default Map and Home Submap

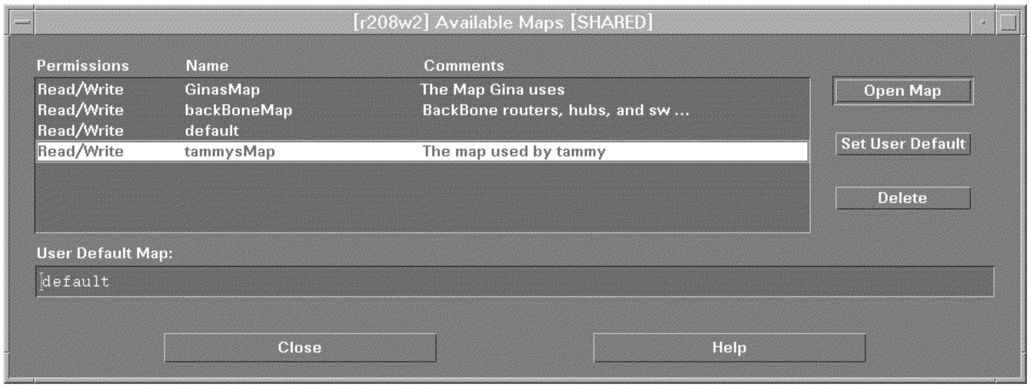
The *user default map* is the map that is open when a user starts the NNM GUI. If the specification of the default map is not modified, the map named “default” will be opened. The administrator can create as many maps as necessary, limited only by disk space.[1] Generally speaking, you

[1] Maps are written to the directory $OV_DB/openview/mapdb. For UNIX systems, use du –k $OV_DB/openview/mapdb to view the sizes of the map databases in kilobytes. For information on the **du** command, refer to the UNIX man page on **du**. For Window systems, view the directory properties from Microsoft Explorer.

should never make customizations to the default map. A better practice would be to make a copy of the default map, saving it under a new name. Customizations should then be made to this new map. After making customizations to a map (such as background graphics, auto-layout, and so on), you may want to set this map as your user default map. The user can set his default map by performing the following:

1. Select from the menu bar **Map→Open…**
2. Select the map you would like as your default (see Figure 6-1).
3. Click the **[Set User Default]** button.

Figure 6-1 Select **Map→Open** from the menu bar to set the user default map. Select the map name and click the **[Set User Default]** button. The User Default Map shown here is the map named "default."



[r208w2] Available Maps [SHARED]

Permissions	Name	Comments
Read/Write	GinasMap	The Map Gina uses
Read/Write	backBoneMap	BackBone routers, hubs, and sw ...
Read/Write	default	
Read/Write	tammysMap	The map used by tammy

User Default Map: default

When a map is opened, the first submap to be displayed is the *home submap*. The home submap is also easily accessible by selecting the "house" icon on the toolbar. By default, the home submap is set to the root submap. The home submap should be set to the submap you access most frequently. Each map may have a different home submap. To set the home submap, navigate to your favorite submap and select from the menubar **Map→Submap→Set This Submap as Home**.

You can now use the toolbar icon to return to the home submap at any time. A user may have as many maps as desired and each map can have a different home submap.

Unused maps begin to contain stale topology data as the NNM topology database is updated over time. Even when **ovtopofix** and **ovw –mapcount** are run, you may encounter errors, such as "objects still in 4 maps." In order to correct this error, each map must be opened so that it synchronizes the current topology information into the map. Each map can either be opened individually or with a script run as root on the NNM server, depending on the number of maps you need to open to synchronize.

Don’t open maps in NNM that are created by OVO! The following example “greps out” two default OVO maps:

```
export DISPLAY=displayhostname:0.0
for  map in ‘ovwls –x | egrep –v “opcadm|opcop”  `
do
  ovw –map $map &
done
```

Check to ensure that every map was opened with read/write privileges. The map will not synchronize without being opened with read/write privileges. After the synchronization of the maps, run **ovw –mapcount**. This should run without errors.

6.1.2 Submap Overlay, Window Geometry, and Symbol Hiding

Many customizations may be made to a map, including the behavior of the submap overlay. By default, when you drill down through a map’s hierarchy by double clicking, the child submap of the selected object replaces the current submap.[2] Likewise, when you click the up arrow icon from the toolbar, the parent submap of the current submap replaces the child. This prevents the user from having too many open submaps. If you prefer not to have this submap overlay feature turned on, you can modify this behavior on both a map and a submap basis by selecting from the menu bar

View→Submap Overlay→Off For This Submap

View→Submap Overlay→Off For All Submaps

Another user customization is the ability to save window size and placement. For instance, if you prefer the IP Internet submap on the left side of your monitor and a particular node-level submap on the right side, simply adjust the submap overlay to open multiple submaps, arrange the submap size and location to your liking, and save the window geometry by selecting **View→Window Geometry→Save For All Open Submaps**.

Further customizations can be made for personal preferences. Not all NNM users want to see the same view of the network. For example, a network manager may never want to see individual workstations or servers. By hiding symbols you can alleviate unwanted clutter from her view of the network. Hidden objects are still actively polled but their status will not be reflected on the maps in which they are hidden. If the network manager hides non-networking devices, she will not see when hidden devices are down by simply glancing at the appropriate submap. However, hiding a device does not prevent the **Node Down Event** from being sent to the Alarm Browser.

[2] In earlier implementations of NNM, every time you double-clicked an object a new window opened. If you double-clicked twenty objects, you had twenty windows. This became a bit cumbersome to keep track of all the open windows. The submap overlay feature allows you to customize this per map and per submap.

To hide symbols, first select them[3] and then select from the menu bar **View→Hidden Objects→Hide Selected From This Submap**.

The status line at the bottom of each submap displays the number of hidden objects on the current submap, as shown in Figure 6-2. You may have a need to hide a particular type of object. For example, you may want to hide all Windows systems because they are all DHCP devices that appear to go down frequently. Follow these steps to hide all systems that have a common attribute such as the operating system:

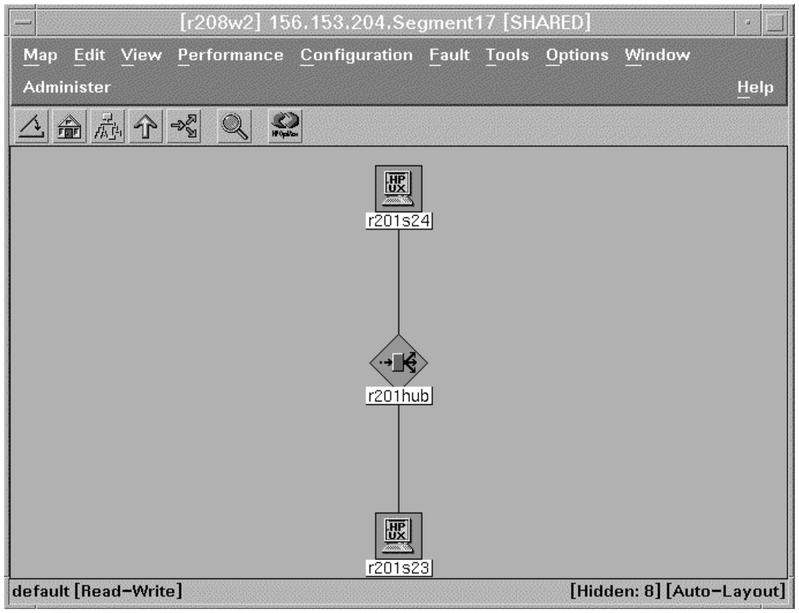
1. Search for the object by the attribute.
   Edit→Find→Object By Symbol Type…
2. Select **Computer** class **Windows NT** subclass and click the **[Apply]** button.
3. Double-click one of the items listed under the "Located and Highlighted" section.
4. Select the items that were found.
   View→Highlights→Select Highlighted
5. Hide the objects that were selected.
   View→Hidden Objects→Hide Selected From This Submap
6. Hidden objects can be unhidden at any time by selecting
   View→Hidden Objects→Show Hidden On This Submap
7. Recall that the submap hierarchy is
   Root→Internet→Network→Segment→Node

Notice that the list includes the segment submap (objects) as well as node submaps. To more quickly and effectively hide the desired objects (in this case, windows NT nodes) we should concentrate on the segment submaps. When you double-click one of these, you are presented with the appropriate child submap which has all NT nodes "highlighted' (highlighted labels are displayed in reverse video).

While hiding objects is one way to deal with unwanted devices, filtering is a more elegant method of dealing with "undesirable" objects. A filter can be applied to on a per-map basis, which hides objects from the view while allowing them to exist in the object database and be actively polled. This type of filter is known as a *map filter*. While the result of applying a map filter is similar to hiding objects, the implementation is usually much faster and more effective. Other types of filters include *topology* and *discovery* filters, which are discussed in detail in Chapter 9, "Scalability and Distribution."

[3] Use the control and left mouse button simultaneously to multi-select symbols. This also toggles the selection.

Figure 6-2 The status line displays the map name (default), the access mode (Read-Write), the number of hidden symbols (Hidden:8), and whether Auto-Layout is enabled.


[r208w2] 156.153.204.Segment17 [SHARED]

default [Read−Write]

[Hidden: 8] [Auto−Layout]

6.1.3 Setting Background Graphics and Automatic Layout

Background graphics can be used to enhance the appearance of a submap. They can be used to display geographic regions, floor plans, or organizational structures. A number of file formats are supported for submap background graphics. If you have a file that is not in the list, check the Release Notes for currently supported file formats. Supported file formats for background graphics are listed in Table 6-1.

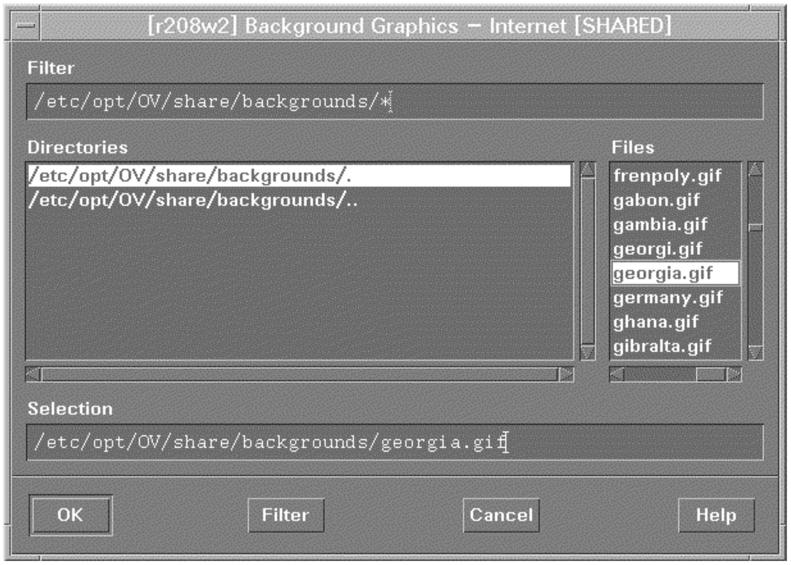
Table 6-1 File types supported by NNM for submap background graphics.

GIF
TIFF
BMP
XBM
PCX
Image
Starbase
XPM
XWD
JPEG

Background graphics may be configured for a submap by following these steps:

1. Select **Map→Submap→Properties** from the menu bar.
2. Click the **[Browse…]** button listed by the Background Graphics section of the dialog box. This is located on the View tab for Windows.
3. Select a graphic as shown in the list of Figure 6-3.
4. Click **[OK]**.

Figure 6-3 The administrator can set a background graphic for each submap. Many of the provided graphics include geographic locations. The administrator can also import a custom graphic such as a floor plan.



[r208w2] Background Graphics – Internet [SHARED]

Filter

/etc/opt/OV/share/backgrounds/*

Directories

/etc/opt/OV/share/backgrounds/.
/etc/opt/OV/share/backgrounds/..

Files

frenpoly.gif
gabon.gif
gambia.gif
georgi.gif
georgia.gif
germany.gif
ghana.gif
gibralta.gif

Selection

/etc/opt/OV/share/backgrounds/georgia.gif

6.1.3.1 Memory Considerations when using Background Graphics

One thing to remember when setting background graphics is the amount of physical memory (RAM) you have on your NNM system. When you open a map, all objects, including background graphics, are loaded into memory[4]. The more objects in the database, the more memory required to load objects into memory. Similarly, the more complex the graphic, the more memory required. One way to tell if your system is memory-bound is if the synchronization process takes a long time (more than a few minutes) when you open a map. If synchronization is taking too long, try decreasing the number of objects being loaded into memory. One way to achieve this is by changing the persistence level of your map.

[4] The default behavior of NNM on UNIX systems is to load all objects in to memory. The default behavior on Windows NT/2000 is to only load objects into memory on the IP Internet submap and higher.

6.1.3.2 Submap Persistence

The persistence level of a submap indicates the level at which objects are loaded into memory. If you select any choice other than **All Levels**, the objects appearing on submaps below that level are not loaded into memory until you open that specific submap. You can define the persistence level of a map by following these steps.

1. Select the map properties.
   Map→Properties
2. Select IP Map and click **[Configure For This Map…]**
3. Scroll down to the section that reads
   On-Demand: To what level should submaps be persistent.
4. Select one of the levels:
   All Levels
   Segment Level and Higher
   Network Level and Higher
   Internet Level
5. After selecting the persistence level, click **[Verify]** and **[OK]** in the IP Map configuration dialog box and **[OK]** in the Map Properties dialog box.

Tip: Any modifications you make to a submap will cause it to become a persistent submap, regardless of the persistence setting. For example, if you set your map persistence level to the **Segment Level and Higher**, only the root, Internet, and network level objects will be loaded into memory. However, if you make any of the following modifications to a segment or node level submap, you will cause that submap to become persistent:

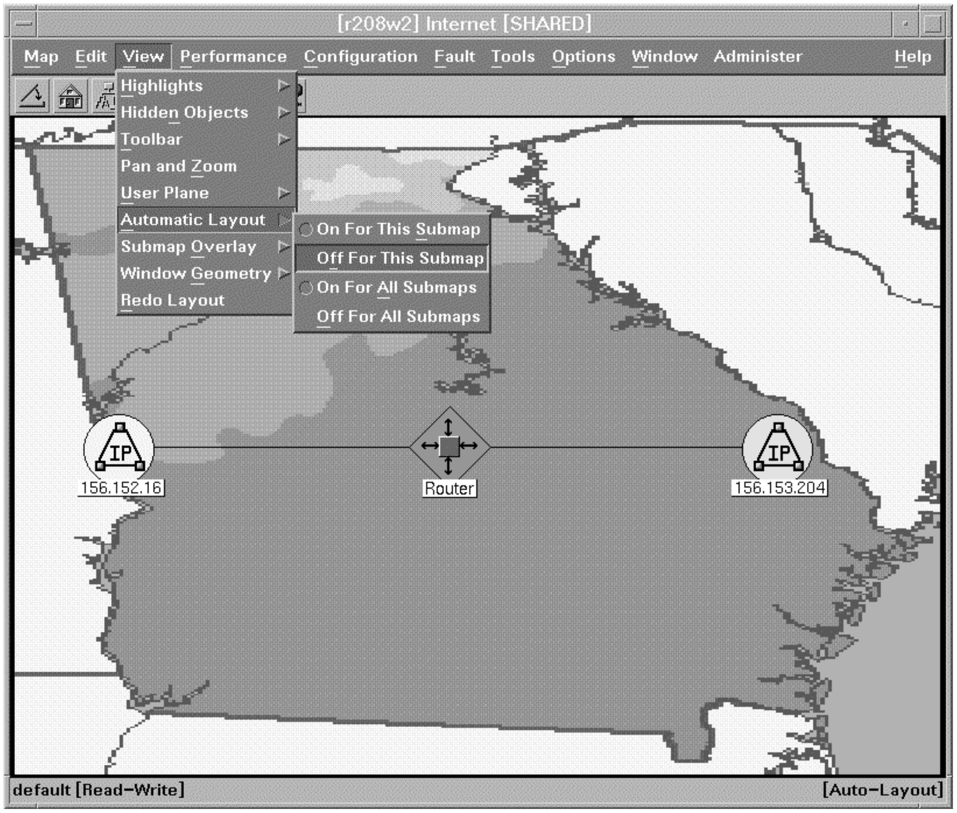
- Submap Overlay
- Window Geometry
- Hidden Objects
- Background Graphics
- Auto-layout

Any objects contained in the submaps that have customizations such as these will be loaded into memory whenever the map is open. As another example, if you set your persistence level to **Internet Level** and set a background graphic on every submap, you defeat the purpose of setting the persistence level.

6.1.3.3 Automatic Layout

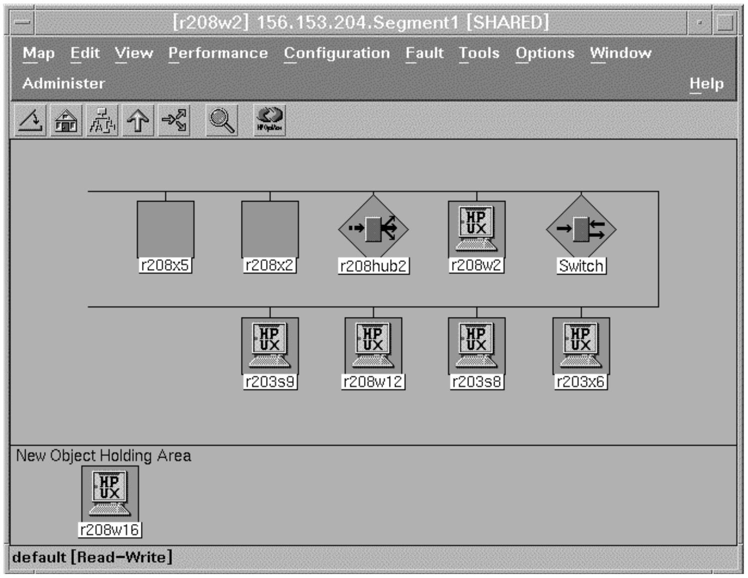
The purpose of automatic layout is to allow a device discovered by NNM to have its symbols placed on a submap without human intervention. When using background graphics, you may find that you would like to manually control the placement of symbols. When placing objects on a floor plan, for example, you can turn off auto layout and place the symbols where they physically reside. To turn off automatic layout for an individual submap, select **View→Automatic Layout→Off For This Submap** as shown in Figure 6-4.

Figure 6-4 This submap has been configured with Georgia as the background graphic. NNM ships with a GIF file for each US state and many countries. Turn off automatic layout to allow manual placement of symbols.



After you turn auto layout off, notice that **[Auto-Layout]** is removed from the status line and a "**New Object Holding Area**" appears at the bottom of the submap, as shown in Figure 6-5. When new devices are discovered that belong on a submap where auto layout is turned off, the newly discovered objects are placed in the New Object Holding Area. You must drag the new objects from the New Object Holding Area and drop them onto the desired location on the submap.

Figure 6-5 The **New Object Holding Area** is created when automatic layout is turned off. Newly discovered devices are placed in the New Object Holding Area. They can be manually placed on a submap by drag-and-drop.



6.1.4 Partitioning the Internet Submap

NNM knows how to lay out devices based on the information provided by SNMP queries. The submap hierarchy from top to bottom is

Root
Internet
Network
Segment
Node

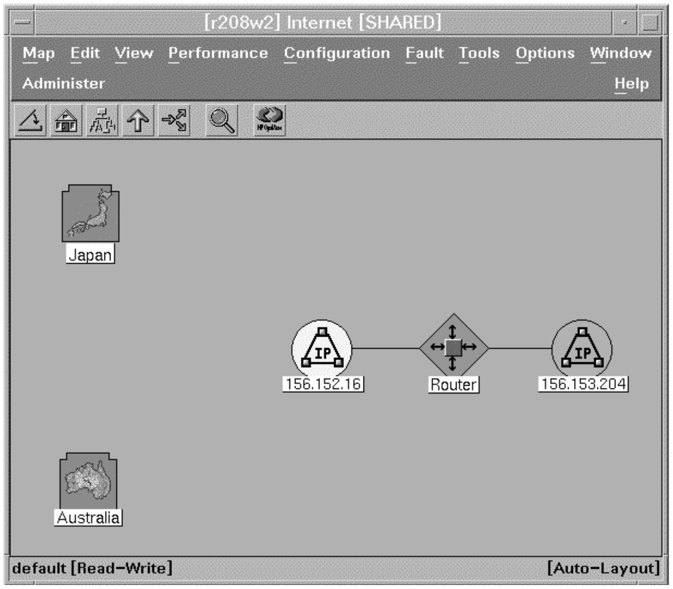
The layout of your network can be customized at the Internet level submap. The purpose is to allow partitioning of your network based on geographical or logical information. NNM does an excellent job of providing the physical connections of your network but makes no assumptions as to the organizational structure. This customization is known as partitioning the Internet submap.

Partitioning the Internet submap requires that you add a special type of symbol called a location symbol. The steps to partition the IP Internet submap are as follows:

1. From the Internet submap select
   Edit→Add Object
2. Select the Location symbol (Figure 6-6) and drag any of the subclass symbols to the submap.
3. Provide a Label and click **[OK].**
4. Repeat the creation of location icons until you have defined all necessary locations.
5. Drag each symbol and drop it in the desired location.[5] Connections will automatically be established between location symbols as shown in Figure 6-7.

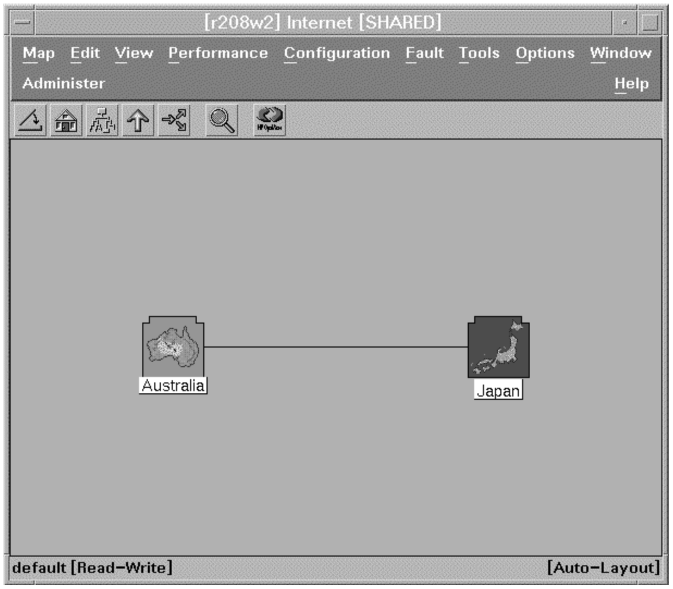
Note: Location icons can be cascaded when partitioning the Internet submap. For example, you could add USA at the Internet level, add State under USA, and add City under State, and then add Building, Site, Room, and so on.

Figure 6-6 Location objects can be used to partition the IP Internet submap. Notice the location objects added to the submap shown (Japan and Australia) are in the application plane (no shading appears around the symbol).



[5] The Windows version of NNM requires using cut and paste instead of drag and drop.

Figure 6-7 When partitioning the Internet submap, connections between the locations are automatically established as symbols are moved into the location icons.



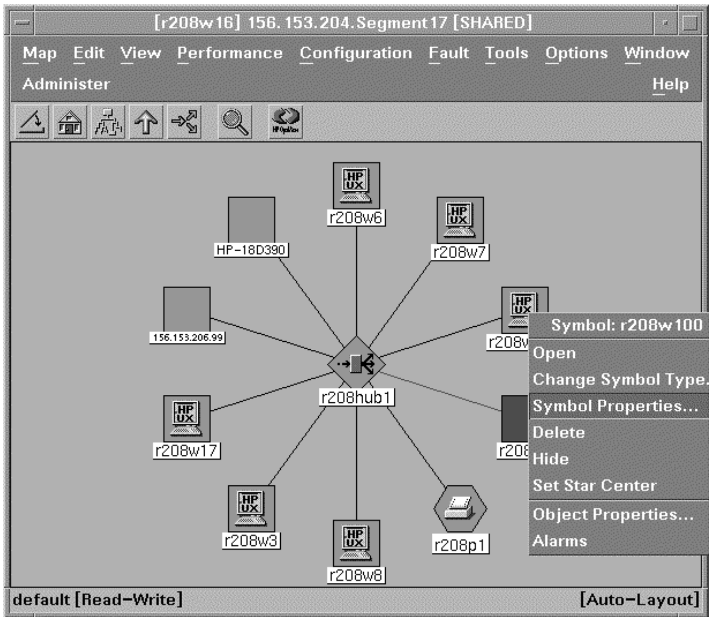
6.1.5 Creating Executable Symbols

By default, symbols on a submap spawn a child submap when double clicked. This behavior is referred to as “explodable.” Symbol properties on a submap may be modified to launch an application when double-clicked rather than opening a child submap. This behavior is referred to as “executable.” A number of pre-defined applications exist and are accessible in NNM. In fact, most applications available from the menu bar and the toolbar may be used in creating executable symbols. These applications are defined in NNM through *Application Registration Files* (ARFs). In addition to using the pre-defined applications, you can create your own applications by defining ARFs as discussed in Chapter 7, “Advanced Customization.”

To make use of the executable symbol feature in NNM, follow the steps listed here. For this example, use a system that allows incoming telnet sessions.

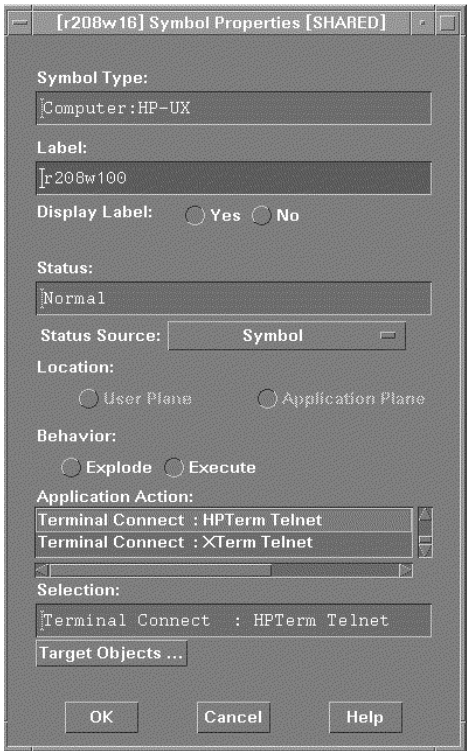
1. Access the symbol properties by right-clicking and selecting **Symbol Properties** as shown in Figure 6-8.

Figure 6-8 The first step in making an icon executable is to place the cursor over the symbol, click the right mouse button, and select **Symbol Properties…** from the symbol popup menu.



2. Modify the symbol behavior to **Execute**, as shown in Figure 6-9. Notice the list of Applications that appear under the section labeled "Application Action."
3. Scroll through the list and select the application
   Terminal Connect: XTerm Telnet
4. Click the **[Target Objects…]** button.
5. Click the **[Add]** button to include the hostname and click **[OK]** on the Target Objects dialog box as shown in Figure 6-10.
6. Click **[OK]** on the Symbol Properties dialog box and you will see the executable symbol as shown in Figure 6-11. When the symbol is double clicked it launches a telnet session to the system.

Figure 6-9 The second step in making an icon executable is to modify the behavior of the symbol to **Execute** and select to application to be launched.



[r208w16] Symbol Properties [SHARED]

Symbol Type: Computer:HP-UX

Label: r208w100

Display Label: ○ Yes ○ No

Status: Normal

Status Source: Symbol

Location: ○ User Plane ○ Application Plane

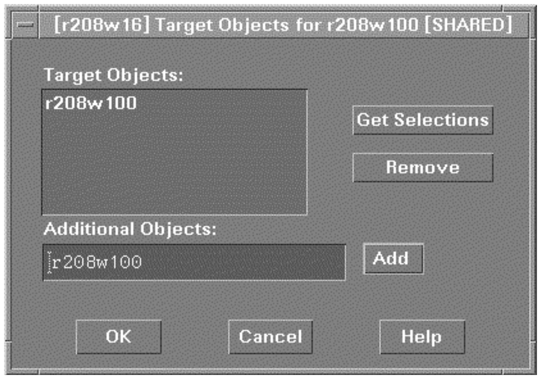
Behavior: ○ Explode ○ Execute

Application Action:

```
Terminal Connect  : HPTerm Telnet
Terminal Connect  : XTerm Telnet
```

Selection: Terminal Connect : HPTerm Telnet

Figure 6-10 The third step in creating an executable icon is to click the **[Target Objects]** button, which displays the dialog box shown. Select the **[Add]** button to add the hostname as a target object.

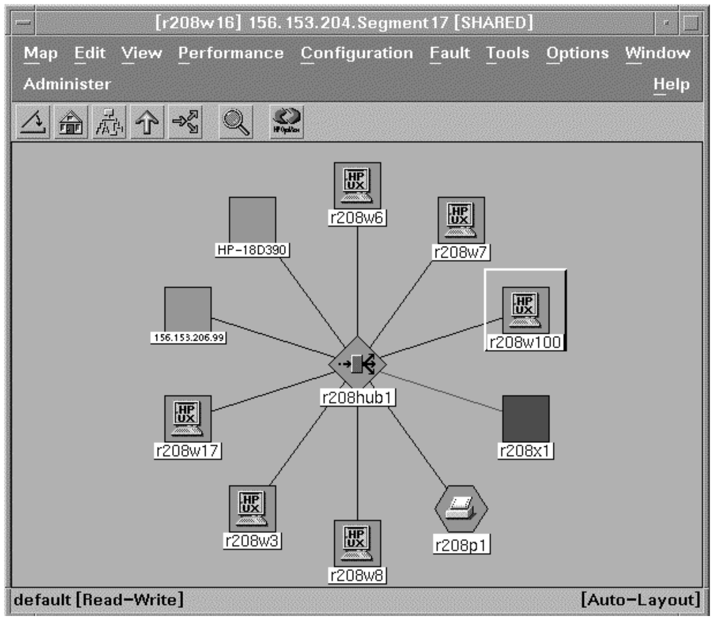


[r208w16] Target Objects for r208w100 [SHARED]

Target Objects: r208w100

Additional Objects: r208w100

Figure 6-11 The executable icon (r208w100) shows up as a button. Double-click it to launch the application. You can also drag a symbol and drop it on the button to launch the application on another object.



After you have created an executable symbol, you can drag other symbols and drop them onto the executable symbol. For example, if you want to launch telnet on a different system, simply drag the symbol and drop it onto the executable symbol. Making a symbol "executable" does not affect the symbol status set by ipmap. There is one caveat when changing an existing symbol's behavior to executable. After changing an existing symbol to executable, you will no longer be able to access the symbol's child submap by double clicking the symbol. Instead, the only way to access the child submap of the symbol is to use the submap open feature. You can do this by selecting **Map→Submap→Open…** You will need to drill down through the submap listings to find the modified symbol's submap. Select the modified symbol and click the **[Open Submaps]** button. For this reason, you can also create standalone symbols that launch applications. It is not necessary to modify the behavior of an existing symbol.

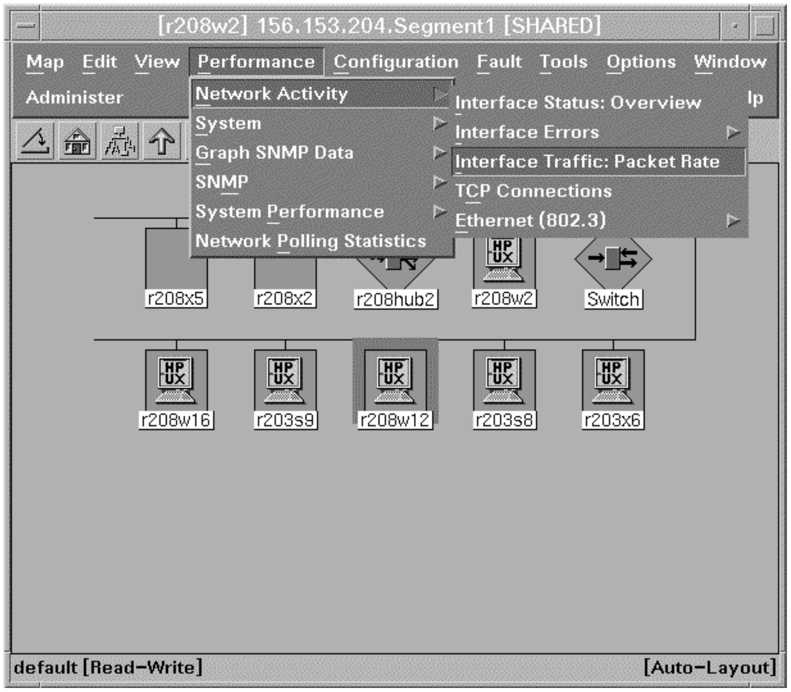
6.1.6 MIB Applications

Many of the options from the tool and menu bar access applications that communicate with the MIB. When launched, these applications send SNMP queries to the selected node to access certain MIB variables. Some applications create tables that contain the data and others generate real-time graphs. NNM provides many built-in MIB applications as well as the capability to create your own MIB applications via the GUI.

6.1.5.1 Built-in MIB Applications

Most MIB applications require you to select a node before launching. If nothing is selected, the menu items will be grayed out and unavailable. Let's look at some of the built-in MIB applications. Select a node and select from the menu bar **Performance→Network Activity→Interface Traffic: Packet Rate** as shown in Figure 6-12. This will display a real-time graph of the MIB variables for packet rate.

Figure 6-12 Interface Traffic is a built-in MIB application available from the menu bar. Select a node and select **Network Activity→Interface Traffic: Packet Rate** from the menu bar. This initiates snmpgets for specific MIB variables and displays them graphically.



6.1.5.2 Creating MIB Applications

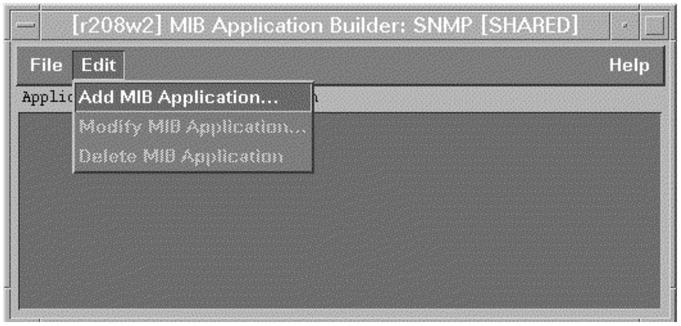
NNM provides the ability to create MIB applications that will be incorporated into the NNM menu bar. This can be achieved by using the MIB application builder.

The MIB application builder allows you to create custom graphs and tables by defining the MIB variables you want to be displayed.

Follow these steps to create your MIB graph:

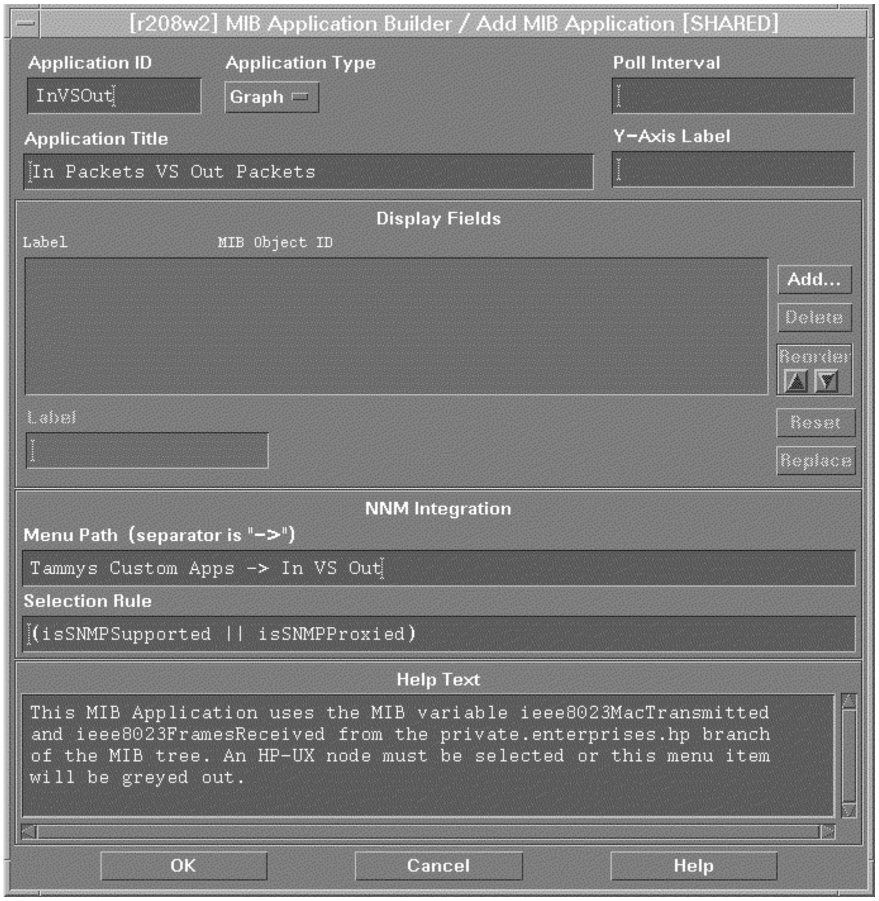
1. Select **Options→MIB Application Builder: SNMP** from the menu bar. This displays an empty MIB application dialog box as shown in Figure 6-13.

Figure 6-13 To create a MIB application, select **Options→MIB Application Builder: SNMP** from the menu bar. The list will be empty by default. Select **Edit?Add MIB Application…**.


[r208w2] MIB Application Builder: SNMP [SHARED]
File
Edit
Help
Applic
Add MIB Application...
Modify MIB Application...
Delete MIB Application

2. Select **Edit→Add MIB Application…**. You will see the dialog box as shown in Figure 6-14.
3. Complete the following fields to define the MIB application:
   Application ID—This must be a unique ID not containing spaces or special characters. For this example, use **InVSOut**
   Application Type—Set this to table, form, or graph. For this example specify graph.
   Application Title—Name the application whatever you want. This can contain spaces and special characters. For this example, specify “In Packets VS Out Packets”.
   Menu Path—This describes the cascaded menu bar for the application beginning with the top-level menu. This may also contain spaces. Use “->” to indicate cascading menus. For this example, specify *yourname* Custom Apps→In VS Out
   Selection Rule—The Selection Rule describes the node type for which this application will work. Until you are familiar with the filtering language described in Chapter 9, it is best to leave it set to the default. The default Selection Rule **(isSNMPSupported || isSNMPProxied)** will function on any device that supports SNMP or has an SNMP proxy.
4. Click the **[Add…]** button to define the MIB variables for the application.

Figure 6-14 Specify the Application ID, Application Type, Application Title, Menu Path and click the **[Add…]** button.



[r208w2] MIB Application Builder / Add MIB Application [SHARED]

Application ID: InVSOut

Application Type: Graph

Poll Interval:

Application Title: In Packets VS Out Packets

Y-Axis Label:

Menu Path (separator is "->"): Tammys Custom Apps -> In VS Out

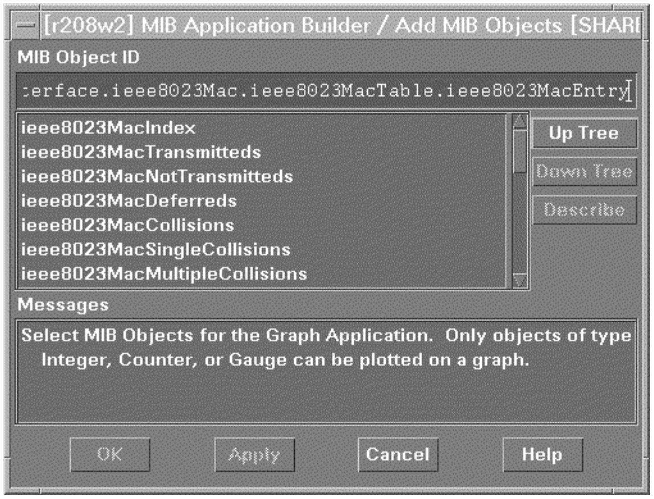
Selection Rule: (isSNMPSupported || isSNMPProxied)

Help Text:

```
This MIB Application uses the MIB variable ieee8023MacTransmitted
and ieee8023FramesReceived from the private.enterprises.hp branch
of the MIB tree. An HP-UX node must be selected or this menu item
will be greyed out.
```

5. Traverse the MIB tree to include the MIB variables for the application. Use the control key and left mouse click simultaneously to select the desired variables. This example uses the MIB variables **ieeeMacTransmitted** and **ieeeMacFramesReceived** shown in Figure 6-15 under the MIB subtree private.enterprises.hp.nm.interface .ieee8023Mac.ieee8023MacTable.ieeeMacEntry
6. After selecting the desired variables, click **[OK]** from the Add MIB Objects dialog box.
7. Click **[OK]** from the Add MIB Application dialog box. You should see your custom menu as shown in Figure 6-16. You are now ready to execute your application.

Figure 6-15 You should be familiar with the MIB tree structure in order to create MIB applications. Press the control key and left click simultaneously to select multiple variables.


[r208w2] MIB Application Builder / Add MIB Objects [SHARI

MIB Object ID

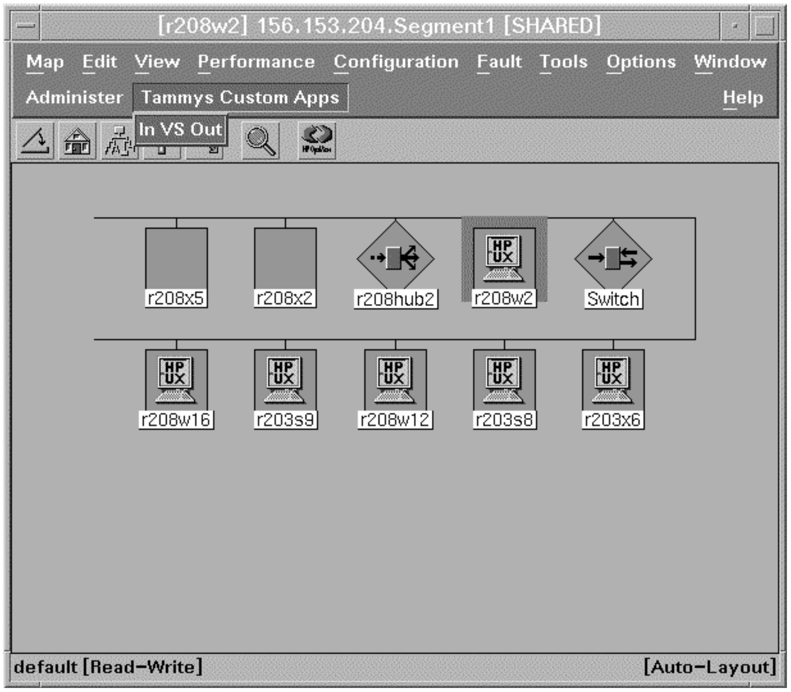
:erface.ieee8023Mac.ieee8023MacTable.ieee8023MacEntry

ieee8023MacIndex
ieee8023MacTransmitteds
ieee8023MacNotTransmitteds
ieee8023MacDeferreds
ieee8023MacCollisions
ieee8023MacSingleCollisions
ieee8023MacMultipleCollisions

Messages

Select MIB Objects for the Graph Application. Only objects of type Integer, Counter, or Gauge can be plotted on a graph.

Figure 6-16 Select the node on which to execute the newly created MIB application. Select the application from the menu bar specified in the Menu Path of the MIB application builder.



6.1.5.3 Use Your MIB Application as an Executable Symbol

After you've created an MIB application it becomes an NNM registered application. The MIB application can now be used when creating executable symbols. Add an object to a submap, change the behavior to executable, and check the list for your application. An *Application Registration File* (ARF) is created when you define an MIB application.

6.2 SUMMARY

Customizing the ovw GUI is both flexible and simple. Many customizations, such as adding submap backgrounds, automatic layout of symbols, symbol hiding, and creation of executable symbols, can be achieved from within the ovw GUI. These customizations allow each NNM user to tailor the environment based on job roles and responsibilities. Submaps are more easily managed, devices are more easily located, and frequently used applications can be easily executed with a double click (when executable symbols are defined). By allowing such customization, the NNM user can become more efficient at managing the network.