

Solstice Site/SunNet/Domain Manager Administration Guide

Copyright 1996 Sun Microsystems, Inc., 2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Portions of this product may be derived from the UNIX[®] system, licensed from Novell, Inc., and from the Berkeley 4.3 BSD system, licensed from the University of California. UNIX is a registered trademark in the United States and other countries and is exclusively licensed by X/Open Company Ltd. Third-party software, including font technology in this product, is protected by copyright and licensed from Sun's suppliers.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19.

Sun, Sun Microsystems, the Sun logo, Solaris, Solstice, Solstice Site Manager, Solstice SunNet Manager, Solstice Domain Manager, and Cooperative Consoles are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the United States and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK[®] and Sun[™] Graphical User Interfaces were developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

X Window System is a trademark of X Consortium, Inc.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.



Contents

1. Overview and Concepts	1-1
1.1 Licensing	1-1
1.2 Site and Domain Differences.....	1-1
1.3 Management Applications and Agents.....	1-2
1.4 SunNet Manager Console	1-5
1.5 Management Database.....	1-6
1.6 Configuration	1-8
2. Planning for Network Management	2-1
2.1 Planning for Network Management	2-1
3. Creating and Modifying the Management Database	3-1
3.1 Adding SunNet Manager to Your PATH Variable	3-1
3.2 Starting the Console.....	3-2
3.3 Using IP Discover and IPX Discover	3-8
3.4 Using IPX Discover	3-11
3.5 Netware Management System Export/Import Agent ...	3-12

3.6	Invoking IPX Discover	3-12
3.7	Traversing the View Hierarchy	3-16
3.8	Creating Elements Using the Editor	3-18
3.9	Creating Aliases	3-24
3.10	Finding Elements	3-26
3.11	Modifying Element Properties	3-27
3.12	Changing Element Types	3-29
3.13	Moving Elements within a View	3-31
3.14	Moving Elements from One View to Another	3-31
3.15	Connecting Elements	3-32
3.16	Copying Elements	3-34
3.17	Deleting Elements	3-36
3.18	Saving the Management Database	3-37
3.19	Quitting the Console	3-39
4.	Requesting Data	4-1
4.1	Making a One-Time Request for Data	4-2
4.2	Making a Request for Periodic Data	4-5
4.3	Prioritizing Requests	4-5
4.4	Copying Requests	4-14
4.5	Viewing Incoming Data	4-15
4.6	Analyzing Stored Data	4-22
4.7	Printing a Graph	4-31
4.8	Viewing and Managing Requests	4-32
4.9	Request States	4-34

4.10 Viewing and Modifying Properties of a Request	4-35
5. Specifying Event Requests	5-1
5.1 Specifying an Event	5-2
5.2 Scheduling Requests Based on Events.	5-2
5.3 Retrieving Single Attributes	5-3
5.4 Detecting the Presence of an Event	5-24
5.5 Checking the Cause of an Event.	5-25
5.6 Changing a Glyph State Back to Normal.	5-27
5.7 Glyph Pending State	5-27
5.8 Propagation of Glyph States	5-29
5.9 Changing the Propagation of Glyph State Changes.	5-30
6. Viewing Reports	6-1
6.1 Viewing Error Messages and Error Reports	6-11
6.2 Viewing Traps	6-15
7. Managing Printers	7-1
8. Managing SNMP Devices	8-1
8.1 Adding SNMP Devices	8-2
8.2 Creating an SNMP Element in the Database.	8-5
8.3 Setting Up SNM to Receive Traps from a Device	8-8
8.4 Using the Set Tool to Retrieve SNMP Attribute Values	8-13
8.5 Using the Set Tool to Change SNMP Attribute Values	8-18
9. Creating and Managing a Link.	9-1
9.1 Using the Console's Edit Function.	9-1
9.2 Using IP Discover to Create Manageable Links	9-10

10. Customizing SunNet Manager	10-1
10.1 Adding Background Image to Current View	10-1
10.2 Creating Types of Elements	10-6
10.3 Creating a New Glyph for an Element Type	10-9
10.4 Modifying the Console Tools Menu	10-10
10.5 Modifying the Tools Menu for an Element Type	10-12
10.6 Adding Agents and Glyphs	10-12
11. Network Management Security	11-1
11.1 Authentication	11-2
11.2 Access Control	11-2
11.3 The Security Algorithm	11-3
11.4 Conferring Right-of-Access	11-4
12. NetWork Layout Assistant	12-1
12.1 Who Should Use NLA?	12-1
12.2 What Does NLA Do?	12-1
12.3 What the Network Layout Assistant Does Not Do	12-2
12.4 Starting the SunNet Manager Console	12-2
12.5 Creating the SunNet Manager Console Database	12-4
12.6 Using the Layout... Option	12-6
12.7 Using the Overview... Option	12-10
12.8 Using the Print... Option	12-14
12.9 The Print Window	12-15
12.10 Tailoring Your Layouts	12-20
12.11 Hierarchical Layout Style	12-20

12.12	Tailoring Your Layout	12-22
12.13	Hierarchical Layout Style	12-22
12.14	Circular Layout Style	12-32
12.15	Symmetric Layout Style.	12-41
12.16	Network Layout Assistant Restrictions.	12-45
13.	Reference Overview	13-1
13.1	Overview	13-1
13.2	Agents and Proxies.	13-2
13.3	SunNet Manager Directories and Files	13-5
13.4	Environment Variables Used with SunNet Manager	13-9
13.5	Extending SunNet Manager	13-12
14.	Console	14-1
14.1	SunNet Manager Console	14-1
14.2	Freezing the Console (Read-Only Mode)	14-4
14.3	Control Panel Buttons and Menus	14-5
14.4	View Button	14-10
14.5	Edit Button	14-11
14.6	Props Button	14-17
14.7	Requests Button	14-17
14.8	Tools Button	14-18
14.9	Goto Button.	14-19
14.10	Element Glyph Menu.	14-20
15.	Requests Management.	15-1
15.1	Send Request.	15-4

15.2	Create Predefined	15-20
15.3	Requests Summary	15-40
15.4	Request Glyph Popup Menu	15-45
16.	View Reports	16-1
16.1	Alarm Reports	16-2
16.2	Data Reports	16-9
16.3	Event/Trap Reports	16-14
16.4	Error Reports	16-16
16.5	Events Summary	16-17
16.6	Add Background	16-18
16.7	Remove Background	16-18
16.8	Find	16-18
16.9	Clipboard	16-19
17.	Props Menu	17-1
17.1	Global Properties	17-2
17.2	Windows	17-5
17.3	Requests	17-6
17.4	Automatic Management	17-7
17.5	Events and Traps	17-16
17.6	Errors	17-22
17.7	Locations	17-25
17.8	Miscellaneous	17-26
17.9	Custom Colors	17-29
17.10	Other Configuration	17-29

17.11	Custom Colors	17-33
18.	Management Database.....	18-1
18.1	Element Type Definition	18-3
18.2	Element Instance Definition	18-9
18.3	Connection Definition	18-11
18.4	Background Definition.....	18-12
18.5	Tools Menu Definition	18-13
18.6	Definition of Requests	18-14
18.7	Duplicate Databases.....	18-23
19.	SNMP Support	19-1
19.1	SNMP Proxy Agent Operation	19-2
19.2	Schema Files	19-8
19.3	SNMP Host Files.....	19-10
19.4	Asynchronous Event Reports (Traps)	19-12
19.5	SNMP Version 2 Support.....	19-27
20.	Browser.....	20-1
20.1	Starting the Browser.....	20-2
20.2	Loading Files.....	20-3
20.3	Report Streams	20-5
20.4	Streams Menu	20-10
20.5	Selecting Streams	20-11
20.6	Folders	20-13
20.7	Customizing the Browser	20-15
21.	Results Grapher.....	21-1

21.1	Results Grapher Window	21-3
21.2	Graph Properties Window.....	21-4
21.3	Displaying Graphs	21-8
21.4	Merging Graphs	21-11
22.	IP Discover.....	22-1
22.1	Invoking IP Discover	22-2
22.2	Discover Tool Configuration.....	22-5
22.3	Updating the Management Database.....	22-19
22.4	snm_discover Command	22-20
22.5	The discover.conf File	22-25
23.	IPX Discover	23-1
23.1	Function Overview.....	23-1
23.2	Forwarding Novell's NMS Alarms to SunNet Manager .	23-4
24.	Set Tool.....	24-1
24.1	Set Tool Window.....	24-3
24.2	Set Information List	24-5
24.3	Invoking Set Tool from the Command Line	24-7
	Glossary.....	Glossary-1
	Index	Index-1

Figures

Figure 1-1	Agent and Proxy Agent Communications Protocol	1-3
Figure 1-2	Using Proxy Agents Across Networks	1-4
Figure 1-3	Using the Console to Initiate Management Tasks and Display Data 1-5	
Figure 2-1	Example of a LAN Configuration	2-3
Figure 2-2	Critical Nodes in Home View	2-4
Figure 2-3	SunNet Manager Console Views	2-5
Figure 2-4	Elements Within Console Views	2-6
Figure 3-1	Quick Start Window	3-3
Figure 3-2	Console Window	3-4
Figure 3-3	IP Discover Home Screen	3-9
Figure 3-4	IP Discover Properties Sheet	3-11
Figure 3-5	IPX Discover Home Screen	3-12
Figure 3-6	IPX Discover Properties Sheet	3-13
Figure 3-7	Selecting a View Name in the Goto Menu	3-17
Figure 3-8	Create Object Window	3-18
Figure 3-9	Object Properties Window (Component)	3-19

Figure 3-10	Top Portions of Object Properties Windows	3-21
Figure 3-11	Object Properties Window	3-23
Figure 3-12	Alias Window.	3-25
Figure 3-13	Properties Window for Router	3-26
Figure 3-14	Selecting the View—Find Menu Item.	3-27
Figure 3-15	Selecting the Glyph—Properties Menu Item.	3-28
Figure 3-16	Element categories for Create Object	3-29
Figure 3-17	Change Type Window	3-30
Figure 3-18	Moving a Glyph to a New Location	3-31
Figure 3-19	Glyph—Connect Example	3-33
Figure 3-20	Selecting the Edit—Copy Menu Item	3-35
Figure 3-21	Selecting the Edit—Delete Menu Item	3-37
Figure 3-22	Selecting the File►Save►Management Database Menu Item	3-38
Figure 3-23	Console Quit Window	3-39
Figure 4-1	Requests Menu—Quick Dump Request	4-2
Figure 4-2	Glyph Menu—Quick Dump Request	4-3
Figure 4-3	Quick Dump Report Window	4-4
Figure 4-4	Request Schedule Menu	4-6
Figure 4-5	Sample Request Builder Window	4-7
Figure 4-6	Request Name Menu.	4-8
Figure 4-7	Sample Data Request Properties Sheet.	4-10
Figure 4-8	Predefined Data Request	4-12
Figure 4-9	View—Data Reports Option	4-16
Figure 4-10	Sample Data Reports Window	4-17
Figure 4-11	Graph Tool Menu	4-18

Figure 4-12	Strip Chart and Indicator Samples	4-19
Figure 4-13	Data Request Properties Template	4-20
Figure 4-14	Strip Chart Menu	4-21
Figure 4-15	Results Browser Window	4-23
Figure 4-16	Results Browser File Menu	4-23
Figure 4-17	Results Browser with Reports Loaded	4-25
Figure 4-18	Browser Edit Menu	4-27
Figure 4-19	Browser Tool Edit Menu	4-28
Figure 4-20	Browser Properties Window	4-28
Figure 4-21	Results Grapher Window	4-30
Figure 4-22	Tools—Snapshot Window	4-31
Figure 4-23	Sample Requests—Summary Window	4-33
Figure 4-24	Requests—Summary Window	4-35
Figure 4-25	Request Glyph Menu	4-36
Figure 4-26	Predefined Data Request Using hostperf Agent	4-38
Figure 4-27	Graph from Predefined Request	4-38
Figure 4-28	Data Request Properties Sheet with Log File Specified	4-40
Figure 4-29	Copying a Request from an Element’s Subview	4-41
Figure 4-30	Data Report Files in Browser	4-42
Figure 4-31	Streams►Graph Grapher Menu Data Report Files	4-43
Figure 4-32	Merged Graphs	4-43
Figure 5-1	Event Request Properties Sheet	5-3
Figure 5-2	Sample Request Builder Window	5-5
Figure 5-3	Request Name Menu	5-6
Figure 5-4	Sample Event Request	5-7

Figure 5-5	Sending Predefined Event Request.	5-9
Figure 5-6	Sample Event Request Properties Sheet w/Blink Glyph Effect	5-10
Figure 5-7	Predefined Event Request Builder	5-11
Figure 5-8	Event Request Properties Sheet w/Color by Priority Glyph Effect 5-12	
Figure 5-9	Accessing snmp system Group for Event Request.	5-13
Figure 5-10	SysUpTime Event Request Properties Sheet	5-14
Figure 5-11	Accessing snmp ifStatus Group for Event Request	5-15
Figure 5-12	Event Request Properties Sheet for ifOperStatus Request . . .	5-16
Figure 5-13	Event Request with traffic agent	5-18
Figure 5-14	Example of traffic Event Request	5-19
Figure 5-15	Managing Log Files: Request Builder Selection	5-21
Figure 5-16	Managing SunNet Manager Log Files: Defining the Event Request 5-22	
Figure 5-17	View—Event/Traps Window	5-26
Figure 5-18	Glyph—Glyph State Menu.	5-29
Figure 5-19	Properties Window for a View	5-30
Figure 5-20	Console—Properties—Category Menu	5-31
Figure 5-21	Properties—Event/Trap Window	5-32
Figure 6-1	Alarm Reports Summary Window.	6-2
Figure 6-2	Alarm Reports Find Window	6-3
Figure 6-3	Alarm Reports: Show View Window	6-4
Figure 6-4	Device-specific Alarm Reports Window	6-6
Figure 6-5	Device-specific Alarm Reports View Menu.	6-7
Figure 6-6	Device-specific Alarm Reports Filter Window	6-8
Figure 6-7	Device-specific Alarm Reports Sort Menu.	6-9

Figure 6-8	Device-specific Alarm Reports Find Window	6-10
Figure 6-9	Save to Logfile Window	6-11
Figure 6-10	Console Window Footer Display	6-12
Figure 6-11	Console Pop-up or Tool Window Footer Display	6-12
Figure 6-12	Sample Error Reports Window	6-13
Figure 6-13	View—Event/Trap Reports Window	6-16
Figure 7-1	Creating a Printer	7-2
Figure 7-2	Properties for New Printer	7-3
Figure 7-3	Sending Predefined Request to Printer	7-4
Figure 8-1	Edit—Create Window	8-5
Figure 8-2	Properties Window for a New Component	8-7
Figure 8-3	Glyph—Set Request Menus	8-14
Figure 8-4	Set Tool Window	8-15
Figure 8-5	Attribute Information Screen	8-16
Figure 8-6	Set Tool—Agent Menu	8-17
Figure 8-7	Set Tool—Group Menu	8-18
Figure 9-1	Creating a Connection Object	9-2
Figure 9-2	Properties for New Connection	9-3
Figure 9-3	New Link in Console	9-4
Figure 9-4	Link in View with Connected Objects	9-5
Figure 9-5	Invoking Send Request to a Link	9-7
Figure 9-6	Request Builder Window	9-8
Figure 9-7	Event Request Properties	9-9
Figure 9-8	Discover Configuration Window	9-10
Figure 9-9	Network Icon for View “Routers”	9-11

Figure 9-10	View Created by Discover	9-12
Figure 9-11	Link Name as Created by IP Discover	9-13
Figure 9-12	Properties of Discovered Link	9-14
Figure 10-1	Add Background File Menu.	10-2
Figure 10-2	Console with New Background	10-3
Figure 10-3	Selecting the File—Save—Management Database Menu Item	10-4
Figure 10-4	File—Save—Management Database Menu	10-5
Figure 10-5	Selecting the File—Load—Management Database Menu Item	10-7
Figure 10-6	File—Load—Management Database Window	10-8
Figure 10-7	Tools—Customize Window.	10-11
Figure 10-8	Properties/Locations Window	10-13
Figure 12-1	Console Window	12-3
Figure 12-2	Overview Window	12-11
Figure 12-3	Print window with default settings	12-15
Figure 12-4	Print window with Secondary Option Panel Displayed.	12-19
Figure 12-5	Main Window - Hierarchical Layout Style Default Settings.	12-21
Figure 12-6	Main Window - Hierarchical Layout Style Default Settings.	12-22
Figure 12-7	Hierarchical Layout, Horizontal Level Orientation (default)	12-23
Figure 12-8	Hierarchical Layout, Vertical Level Orientation.	12-24
Figure 12-9	Hierarchical Layout, 60 Percent Level Frame Spacing (default) 12-25	
Figure 12-10	Hierarchical Layout, 120 Percent Level Frame Spacing.	12-26
Figure 12-11	Hierarchical Layout, 60 Percent Node Frame Spacing (default) 12-27	
Figure 12-12	Hierarchical Layout, 120 Percent Node Frame Spacing	12-28
Figure 12-13	Hierarchical Layout, 0.20 Minimum Slope (default)	12-29

Figure 12-14	Hierarchical Layout, 0.50 Minimum Slope	12-29
Figure 12-15	Hierarchical Layout, Wide Buses	12-30
Figure 12-16	Hierarchical Layout, Narrow Buses (default)	12-31
Figure 12-17	Hierarchical Layout, “As Is” Buses.	12-31
Figure 12-18	Main Window Showing Circular Layout Style Selected	12-32
Figure 12-19	Circular Layout, 60 Percent Node Frame Spacing (Default)	12-33
Figure 12-20	Circular Layout, 120 Percent Node Frame Spacing	12-34
Figure 12-21	Internetwork Before Circular Layout Grouping.	12-35
Figure 12-22	Internet After Circular Layout Grouping	12-35
Figure 12-23	Main Window Showing Symmetric Layout Style Selected.	12-41
Figure 12-24	Symmetric Layout, 60 Percent Node Frame Spacing (Default)	12-42
Figure 12-25	Symmetric Layout, 120 Percent Node Frame Spacing.	12-43
Figure 12-26	Symmetric Layout, Start Seed of 1 (default)	12-44
Figure 12-27	Symmetric Layout, Start Seed of 2	12-45
Figure 14-1	Console Read-Only Mode	14-5
Figure 14-2	File Menu	14-7
Figure 14-3	Load/Save Management Database Window	14-8
Figure 14-4	Edit Button Menu	14-11
Figure 14-5	Edit>Create Menu	14-14
Figure 14-6	Edit>Create >Element Properties Window	14-15
Figure 14-7	Tools Menu.	14-18
Figure 14-8	Sample Goto Menu	14-20
Figure 14-9	Glyph Menu	14-21
Figure 14-10	Glyph Menu—Tools	14-23
Figure 14-11	Glyph Menu—Glyph States.	14-25

Figure 14-12	Sample Show Subview Window	14-26
Figure 14-13	Glyph Menu—Properties—Alias Window	14-30
Figure 14-14	Connection Created Between two Elements	14-31
Figure 14-15	Glyph Menu—Change Type Window	14-32
Figure 14-16	Glyph Menu—Auto Manage Off	14-33
Figure 15-1	Requests Menu	15-2
Figure 15-2	Quick Dump Request	15-3
Figure 15-3	Send Requests Window	15-4
Figure 15-4	Data Request Template	15-5
Figure 15-5	Event Request Template	15-12
Figure 15-6	Sample Predefined Data Request Template	15-24
Figure 15-7	Sample Predefined Event Request Template	15-31
Figure 15-8	Requests Menu—Requests Summary Window	15-41
Figure 15-9	Selecting Requests	15-43
Figure 15-10	Request Glyph Menu	15-45
Figure 15-11	Sample Request Properties Window	15-46
Figure 16-1	View Button Menu	16-2
Figure 16-2	Alarm Reports Summary Window	16-3
Figure 16-3	Device-specific Alarm Reports Sort Menu	16-4
Figure 16-4	Device-specific Alarm Reports Filter Window	16-6
Figure 16-5	Save to Logfile Window	16-7
Figure 16-6	Alarm Reports Show View Window	16-8
Figure 16-7	Alarm Reports Find Window	16-9
Figure 16-8	View -- Data Reports Window	16-10
Figure 16-9	Strip Chart	16-12

Figure 16-10	Strip Chart Properties	16-13
Figure 16-11	View — Event/Trap Reports	16-15
Figure 16-12	Drop All Option Popup Menu	16-17
Figure 16-13	View—Find Window	16-19
Figure 17-1	Selecting Props Button	17-2
Figure 17-2	Console Properties Window	17-3
Figure 17-3	Console Properties Window Categories	17-4
Figure 17-4	Console Properties Requests Category	17-6
Figure 17-5	Console Properties Automatic Management Category	17-8
Figure 17-6	Automatic Management Customize Window	17-13
Figure 17-7	Customizing Auto Management	17-14
Figure 17-8	Console Properties Events and Traps Category	17-16
Figure 17-9	Events and Traps—Upon Opening Menu	17-17
Figure 17-10	Custom Colors Category of Console Props Menu	17-19
Figure 17-11	Custom Color Window	17-20
Figure 17-12	Console Properties Errors Category	17-23
Figure 17-13	Console Properties Locations Category	17-25
Figure 17-14	Console Properties Miscellaneous Category	17-26
Figure 17-15	Custom Colors Window	17-29
Figure 17-16	Custom Colors Configuration Window	17-33
Figure 19-1	SNMP Proxy Agent	19-2
Figure 19-2	MIB and Schema Definition	19-3
Figure 19-3	Sample Properties Sheet for Pseudo-Devices	19-21
Figure 19-4	Trap Report	19-23
Figure 20-1	Results Browser Window	20-3

Figure 20-2	Load Window.	20-4
Figure 20-3	Results Browser Report Streams	20-5
Figure 20-4	Agent Reports from Selected Stream.	20-7
Figure 20-5	Report Menu.	20-8
Figure 20-6	Results Browser Streams Menu.	20-10
Figure 20-7	Streams Selection by System	20-11
Figure 20-8	Sending Data to the Grapher	20-12
Figure 20-9	Copying Streams to a Folder	20-14
Figure 20-10	Tool Properties Window.	20-16
Figure 21-1	Invoking the Grapher from the Console	21-2
Figure 21-2	Results Grapher Window.	21-4
Figure 21-3	Graph Properties	21-5
Figure 21-4	Graph.	21-9
Figure 21-5	Controlling Grapher Rotation Angles	21-10
Figure 22-1	IP Discover Tool Base Window	22-3
Figure 22-2	Discover Configuration Window:.	22-5
Figure 22-3	Subview Created without Coordinates	22-12
Figure 22-4	Example of Routers-Only IP Discover Configuration	22-14
Figure 22-5	Subview in Routers-only Hierarchy	22-15
Figure 22-6	IP Discover Configuration: Monitor Window.	22-16
Figure 22-7	Example of Monitor Configuration	22-20
Figure 23-1	IPX Discover Home Screen	23-3
Figure 23-2	IPX Discover Configuration Screen	23-4
Figure 24-1	Invoking Set Tool.	24-2
Figure 24-2	Set Tool Window	24-3

Figure 24-3	Set Tool Attributes Window	24-5
-------------	----------------------------------	------

Tables

Table 2-1	Summary of Shipped Agents and Proxies	2-8
Table 3-1	Glyphs Used for Multiple Element Types	3-22
Table 4-1	SNM Supplied Predefined Data Requests	4-37
Table 5-1	Predefined Event Requests Supplied with SunNet Manager	5-8
Table 13-1	Agents Specific to Solaris 2.x	13-3
Table 13-2	Summary of Default SNM File Locations	13-10
Table 14-1	Glyphs Used for Multiple Element Types	14-16
Table 14-2	Summary of Edit Operations	14-17
Table 15-1	SNM Supplied Predefined Data Requests	15-22
Table 15-2	SNM Supplied Predefined Event Requests	15-23
Table 18-1	Data Request — dataRequest Record	18-15
Table 18-2	Data Request—dataAttribute Record	18-17
Table 18-3	Data Request — rqstState Record	18-18
Table 18-4	Data Request — membership Record	18-18
Table 18-5	Event Request — eventRequest Record	18-19
Table 18-6	Event Request — eventAttribute Record	18-21

Table 18-7	Event Request — <code>rqstState</code> Record	18-22
Table 18-8	Event Request — <code>membership</code> Record	18-22
Table 19-1	Supported SNMP and Schema Types.	19-22
Table 19-2	Supported SNMP and Schema Types.	19-25
Table 22-1	Sun Machine Types	22-28

Preface

This Administration Guide provides Task and Reference information for the current Solstice Site/SunNet/Domain Manager product (hereafter referred to as SunNet Manager or SNM). This Guide is intended to help you perform basic network management tasks. Part 1, "Tasks," provides steps for performing many common network management tasks. Part 2, "Reference," provides detail on SunNet Manager features, functions and tools.

Like other OpenWindows applications, the SunNet Manager Console uses the OPEN LOOK(tm) graphical user interface. The OPEN LOOK interface allows you to perform a variety of operations through direct manipulation of windows, icons, glyphs and menus on your workstation screen. This Guide assumes you already know how to use the OPEN LOOK screen objects, such as buttons, scroll bars, and the like. If you are new to the Open Windows user interface, you may wish to consult the following SunSoft Press books:

- *Solaris Open Windows User's Guide*
- *John A. Pew, Guide to Solaris*

At the end of this chapter is an explanation of conventions followed in this Guide when describing use of the mouse.

Who Should Use This Book

The document is intended both for first-time and more experienced SunNet Manager users.

Reorganization of Information

Combines User and Reference Guides

This book is new for version 2.3 of SunNet Manager. It combines information that was in the previous *User's Guide* and *Reference Manual*. This new organization is designed to help you access information more easily and efficiently.

Man Pages

The Man Pages are now in a new book titled, *Solstice/Site/SunNet/Domain Manager Reference Guide*.

Error Messages

The Error Messages are now in a new book, titled, *Solstice/Site/SunNet/Domain Manager Troubleshooting Guide*.

Administration Guide Organization

This *Administration Guide* has two Parts. **Part 1: Tasks** describes how to perform various SunNet Manager tasks. **Part 2: Reference** presents reference information on SunNet Manager features and functions.

Part 1: Tasks, is organized as follows:

Chapter 1, "Overview and Concepts," provides a high-level description of SunNet Manager and its underlying architecture.

Chapter 2, "Planning for Network Management," describes the planning you should do before installing SunNet Manager.

Chapter 3, "Creating and Modifying the Management Database," describes how to start and quit the Console, and how to create and modify the runtime database.

Chapter 4, "Requesting Data," describes how to create and send data requests.

Chapter 5, "Specifying Event Requests," describes how to specify conditions that will trigger an event, check for the presence and cause of an event, and change the state of a Glyph.

Chapter 6, “Viewing Reports,” describes how to view event, data, trap, and error reports.

Chapter 7, “Managing Printers,” describes how to manage remote printers.

Chapter 8, “Managing SNMP Devices,” describes how to add SNMP devices and perform other related tasks.

Chapter 9, “Creating and Managing a Link,” describes how to use the Console’s Edit function and how to use IP-Discover to create links.

Chapter 10, “Customizing SunNet Manager,” describes the ways you can modify SunNet Manager to perform tasks beyond sending requests and gathering data.

Chapter 11, “Network Management Security,” describes how to restrict access to agent services.

Chapter 12, “NetWork Layout Assistant,” describes NLA functions and how to use them.

Part 2: Reference, is organized as follows:

Chapter 13, “Reference Overview,” gives an overview of the functions and features discussed in the Reference Part.

Chapter 14, “Console,” describes the Control Panel features and functions.

Chapter 15, “Requests Management,” identifies predefined requests sent with SunNet Manager and gives details on the options available when sending data and event requests.

Chapter 16, “View Reports,” describes the features of the View button.

Chapter 17, “Props Menu,” describes the features of the Console Props button.

Chapter 18, “Management Database,” describes the files that comprise your management database.

Chapter 19, “SNMP Support,” discusses SunNet Manager SNMP support.

Chapter 20, “Browser,” describes the Browser tool.

Chapter 21, “Results Grapher,” describes the Grapher tool.

Chapter 22, “IP Discover,” describes how to use the IP Discover tool.

Chapter 23, “IPX Discover,” describes how to use the IPX Discover tool.

Chapter 24, “Set Tool,” describes how to change attribute values.

“Glossary” provides a glossary of terms used throughout the SunNet Manager documents.

Compatibility

See the *SunNet Manager 2.3 Important Product Information (IPI)* for compatibility information.

Conventions Used in This Book

Command Line Examples

All command line examples in this guide use the C-shell environment. If you use either the Bourne or Korn shells, refer to `sh(1)` and `ksh(1)` man pages for command equivalents to the C-shell.

What Typographic Changes and Symbols Mean

The following table describes the type changes and symbols used in this book.

Table P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. system% You have mail.
AaBbCc123	What you type, contrasted with on-screen computer output	system% su Password:
<AaBbCc123>	Command-line placeholder: replace with a real name or value	To delete a file, type <code>rm</code> <filename>.

Table P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
<i>AaBbCc123</i>	Book titles, new words or terms, or words to be emphasized	These are called <i>class</i> options. You <i>must</i> be root to do this.
Code samples are included in boxes and may display the following:		
%	UNIX C shell prompt	<code>system%</code>
\$	UNIX Bourne and Korn shell prompt	<code>system\$</code>
#	Superuser prompt, all shells	<code>system#</code>

Part 1 — Network Management Tasks

Overview and Concepts



Solstice Site/SunNet/Domain Manager (hereafter referred to as SunNet Manager or SNM) is a comprehensive set of tools and services to help you perform basic network management tasks. SunNet Manager is also an extensible platform that allows you to develop your own network management applications. This chapter introduces basic concepts underlying the use of SunNet Manager.

1.1 Licensing

SunNet Manger is a licensed product. You need a password for each machine on which it is installed. You can obtain a license password from the license distribution center for your region or country. See your Installation Guide for the addresses and phone numbers of license distribution centers.

1.2 Site and Domain Differences

Solstice Site Manager has a license restriction of 100 nodes and includes the sender portion of Cooperative Consoles. The sender portion allows management data (topology, events, and traps) to be forwarded to Domain Manager. All other features of the current release are included in Site Manager. The SNM proxy agent does remote polling and sends the data back to the console using RPC.

Solstice Domain Manager is typically used to manage large site or multi-site networks. Up to 10,000 nodes can be managed, and the sender and receiver portions of Cooperative Consoles are included. In addition to the full suite of SunNet Manager tools, including remote polling by the SNMP proxy agent, Domain Manager can use more than 300 partner applications to augment network management and data analysis.

1.3 Management Applications and Agents

The SunNet Manager product provides both management applications and agent software. You install the SunNet Manager software on the system from which you will manage the network—this system is known as the *management station*. *Management applications* are the processes that allow you to initiate management tasks and collect management information. *Agents* are processes that access the device or element being managed at the request of a management application.

Most agents that are provided with SunNet Manager return information about entities on the Sun workstation on which the agent software is installed. A second type of agent, a *proxy agent*, provides information about entities on *other* systems or other vendors' devices. Each agent returns a certain set of information or *attributes* to the management application. For example, the `hostmem` agent returns information about memory usage on the system on which the agent is installed. The Simple Network Management Protocol (SNMP) proxy agent returns information about SNMP objects on any device that supports the SNMP standard.

Proxy agents provide two main advantages:

- They allow the management application to manage objects using virtually any protocol. SunNet Manager agents and proxy agents communicate with the management applications through the Remote Procedure Call (RPC) protocol. Proxy agents translate the RPC protocol into the protocol that the managed objects understand. This is illustrated in Figure 1-1.

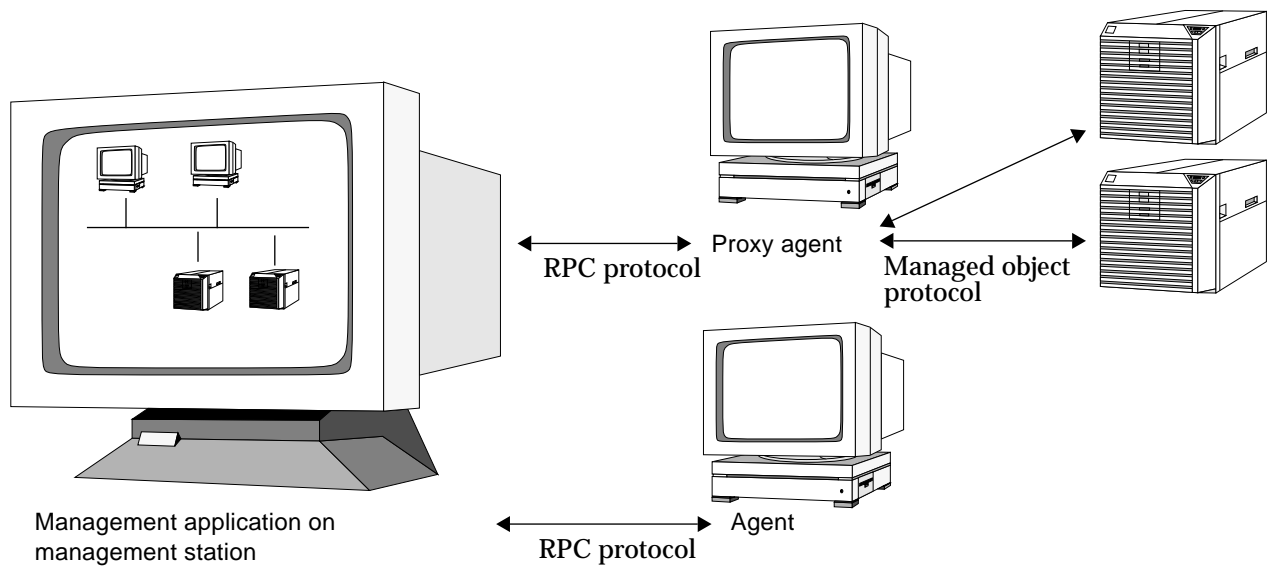


Figure 1-1 Agent and Proxy Agent Communications Protocol

- A single proxy agent can provide management access to multiple devices. The management application only needs to communicate with one proxy agent to manage many devices. The real advantage of this becomes apparent when the proxy agent is installed in a different subnet or domain from the management application. The proxy agent handles the low-level gathering of data from the managed objects. Only minimal network traffic containing the relevant management information passes between the proxy agent and the management application. This is illustrated in Figure 1-2.

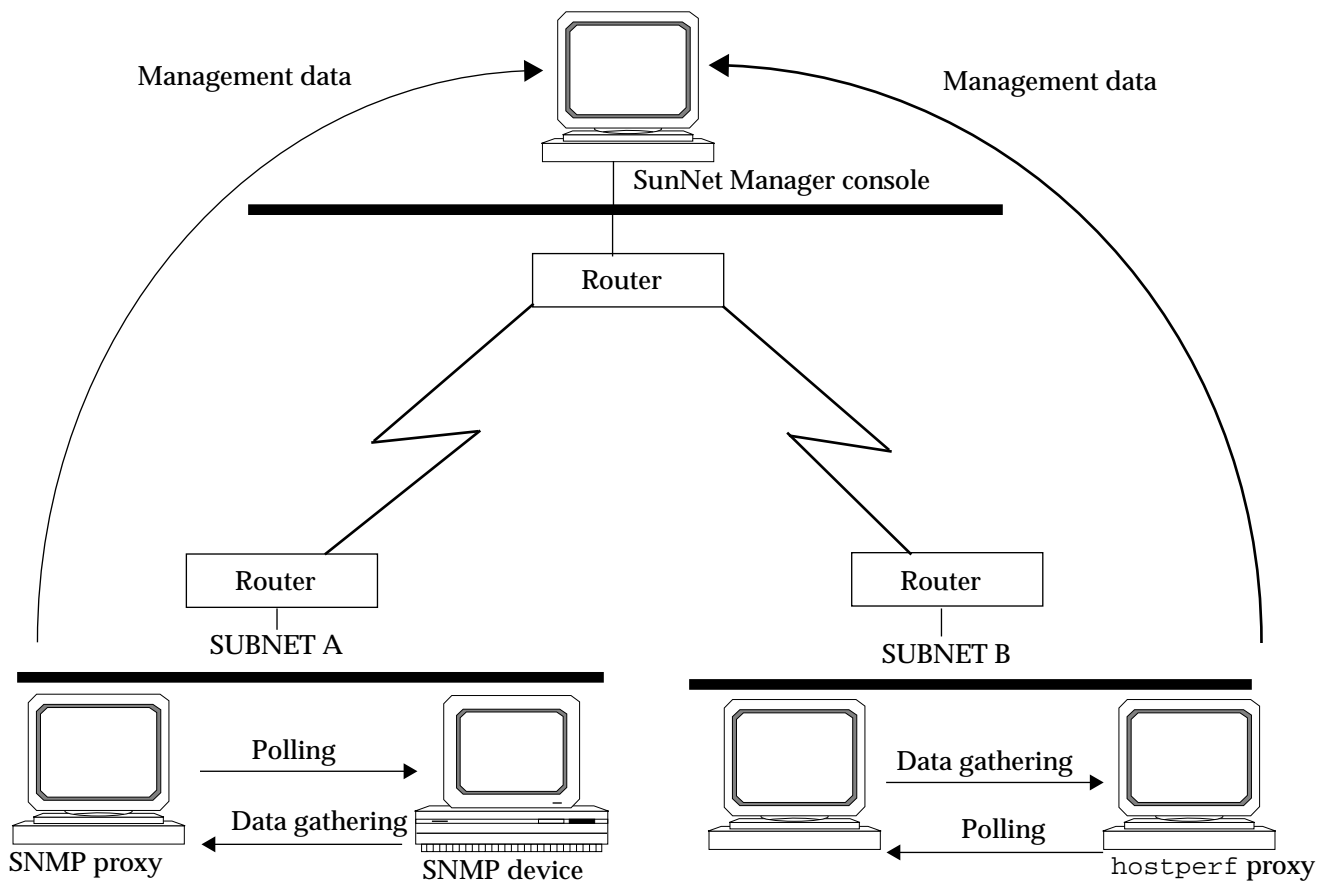


Figure 1-2 Using Proxy Agents Across Networks

The SunNet Manager package includes a collection of agents and proxy agents. For a list of SunNet Manager agents and brief descriptions of the data they return, see “Part 2: Reference.”

1.4 SunNet Manager Console

The SunNet Manager Console is the central management application in the SunNet Manager package. The Console is a graphically-oriented interface that allows you to create a representation of your network. You can use the Console to initiate management tasks and display management information. Figure 1-3 shows some examples of Console functions.

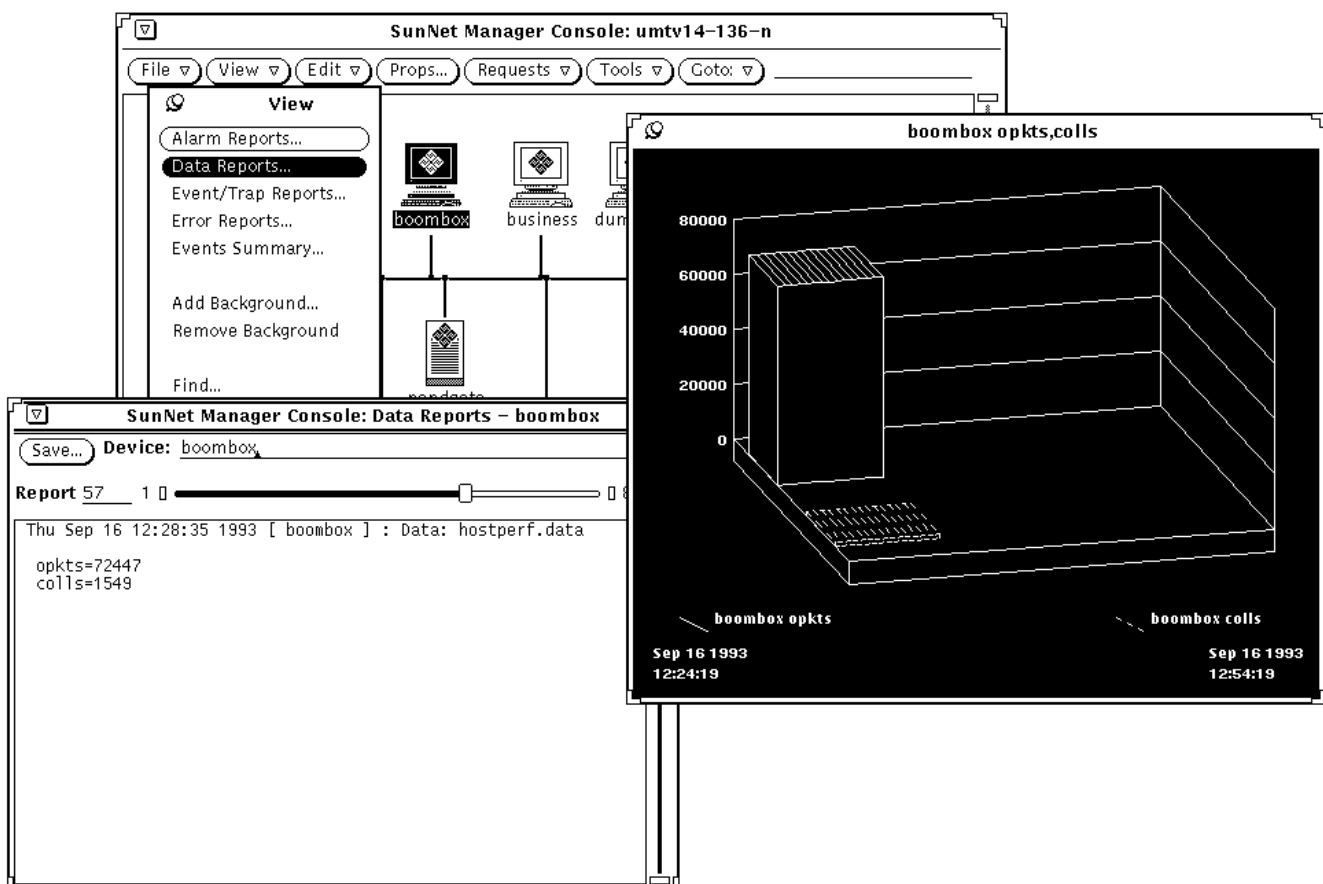


Figure 1-3 Using the Console to Initiate Management Tasks and Display Data

The Console provides mechanisms to initiate requests to agents for *data reporting* and for specifying *events*.



Data reporting allows you to direct agents to send reports of management data on a periodic basis. For example, you can direct the `hostperf` agent to return at one-hour intervals the percentage of CPU being used on a particular system. You can choose to have the reported data displayed in a log, in a chart or graph, or stored in a disk file.

SunNet Manager provides additional tools for viewing and analyzing the returned data: the Results Browser allows you to analyze data that has been stored to a disk file, while the Results Grapher allows you to see a graphical representation of either incoming data or stored data.



Event reporting allows you to direct agents to send reports of management data when an event takes place in the network. An event is an occurrence of certain user-defined conditions. For example, you can direct the `hostperf` agent to send a report whenever the CPU percentage on a system exceeds a set number. You can choose to have the Console reflect the report of an event by visual or audible indicators, or have the report of an event automatically launch a predefined program.

1.5 Management Database

The Console and other management applications rely on a *management database* (MDB) that contains definitions of the elements being managed, the agents that are available, and the requests that have been made to agents. The management database contains:



- **Definitions of each *type* of element that can be represented in the SNM Console.** This element type definition specifies the name of the element type (for example, `ss10` for a SPARCstation 10) and the glyph (or icon) associated with it. The `elements.schema` file provided with SunNet Manager defines many general element types.

The `elements.schema` file is located in the `struct` directory. (The default path for this directory is `/usr/snm/struct` for the Solaris 1.1.1 version of SNM, or `/opt/SUNWconn/snm/struct` for the Solaris 2.x version.) In addition, you can create your own element schema file that defines one or more element types.



doctest

- **Definitions of *instances* of element types.** The element instance definition represents a particular element in your network—usually the name of a device. An element instance also defines those agents that can be used to manage the element. In the Console, glyphs represent the instance definition

of each element. Element instance definitions can be created automatically by a management application such as the Discover Tool, or created “manually” using the Console’s Edit►Create function.

- **Definitions of the agents that the management application can use to manage elements.** Each agent can return different sets of information or *attributes*. The set of attributes that can be returned by each agent is defined in an agent *schema* file. At least one agent schema file should be installed on the management station for each agent that a management application will direct. (The SNMP proxy agent can be used with many agent schema files for different SNMP devices. Three SNMP schema files are included with SNM: `snmp.schema`, `snmp-mibII.schema`, and `sun-snmp.schema`.)

Note that while technically all agents and schema files are available for any element instance, some agents are more appropriate for certain element types than others. A set of agent schema files are provided with SunNet Manager. The default location for these agent schema files is:

- `/usr/snm/agents`, if you’ve installed the Solaris 1.1.1 version of SNM
- `/opt/SUNWconn/snm/agents`, if you’ve installed the Solaris 2.x version of SNM

The contents of the agent schema files are described in the man page for each agent.

- **Definitions of predefined requests that you can invoke for a managed object in the Console.** SNM has a set of predefined requests that cover a wide range of your information-gathering needs. These save you the trouble of building requests for individual elements. These requests are stored in `$HOME/.SNMpredefined`.

The management database present while you are running the Console—referred to as the *runtime database*—can be saved to an ASCII file and later reloaded into the Console. This feature allows you to save or backup your database—with any customizations you might have made—across system reboots. It also allows you the advantage of a portable database file. For example, you can manage multiple databases from the same Console or, within the same network, manage the elements in the same database from different machines.

1.6 Configuration

The two basic types of SunNet Manager configuration are:

- **Configuration of the operation of the Console and SunNet Manager tools.** You change values associated with these programs through the Props (Properties) button in the Console window. An example of this type of configuration is changing the way a glyph responds to an event from blinking (the default) to changing color. See Chapter 17, “Props Menu” in “Part 2: Reference” for a description of configurable properties.
- **Configuration of certain operating characteristics of SunNet Manager agents and daemons on the system on which the agents and daemons are installed.** Most network administrators do not need to perform this type of configuration.

The characteristics of SNM agents and daemons are defined in the `snm.conf` file. For example, you can specify the locations of the log files generated by the SunNet Manager daemons. (The `snm.conf` file is located in the `/etc` directory if you’ve installed this product on a SunOS 4.x machine; if you’ve installed the Solaris 2.x version of this product, this file is located in `/etc/opt/SUNWconn/snm`.)

Additionally, you can specify information relevant to the operation of the SNMP proxy agent, such as the location of SNMP schema files and the maximum number of requests that an SNMP proxy agent subprocess will handle. You can also specify security access for agents on the system where the `snm.conf` file resides. See the `snm.conf (5)` man page and the man pages for individual agents for more information. There is list of all the agents shipped with the current product in Table 2-1 in the next chapter.

Planning for Network Management



This chapter offers ideas on how to set up and use SunNet Manager to meet your network management goals. The methods described here are not the *only* way these goals can be met. Your approach will depend upon your particular network configuration, your network management priorities, and the network management applications you have.

2.1 Planning for Network Management

Before installing the agent software and starting the management Console, you need to plan for the installation. Ask yourself this question: How will we use SunNet Manager to manage our network?

The following steps indicate the types of specific questions you will need to answer:

1. What are our critical nodes?

Identify devices that have impact on the greatest number of network users-- devices such as gateways, hubs, print servers, and software servers. If you only want to monitor these devices, you can reduce the number of elements to create and monitor.

2. What views of the network are most important?

On a small network, you might place all critical nodes into a single view. For example, Figure 2-1 shows a network consisting of two subnets, A and B, which are connected by a gateway.

You could create a separate view for each subnet as well as separate views by type of device — routers, software servers, and print servers. You could create these views one-at-a-time using the Console Edit►Create function, as described in Section 3.8, “Creating Elements Using the Editor.” Usually, it is more convenient to let the Discover Tool build a hierarchy of views to represent your network topology, as described in the Tools section of this manual. (You should use IPX Discover, if you have Novell Netware IPX nodes in your network.) For the example in Figure 2-1, you could create views in the Home view which would group elements by function (software servers, routers, etc.) and subnetwork.

SPARCstations and diskless workstations mount software from servers. One software server in each subnet doubles as a print server. An IPX.25 router provides a connection to a public switching network.

Assuming the servers and gateways are critical nodes, they could be placed into a single view. You could create these elements one-at-a-time, using the Console’s Edit►Create function, as described in Section 3.8, “Creating Elements Using the Editor.” With only one view required, the elements could be placed in the Home view, as shown in Figure 2-2.

For most situations, however, you will want multiple views to represent functional groupings of network devices and to represent the network topology — the various networks and subnetworks, types of connections used, and locations of routers and gateways. For example, you might want views to depict devices in particular buildings, or a view that consists of only routers.

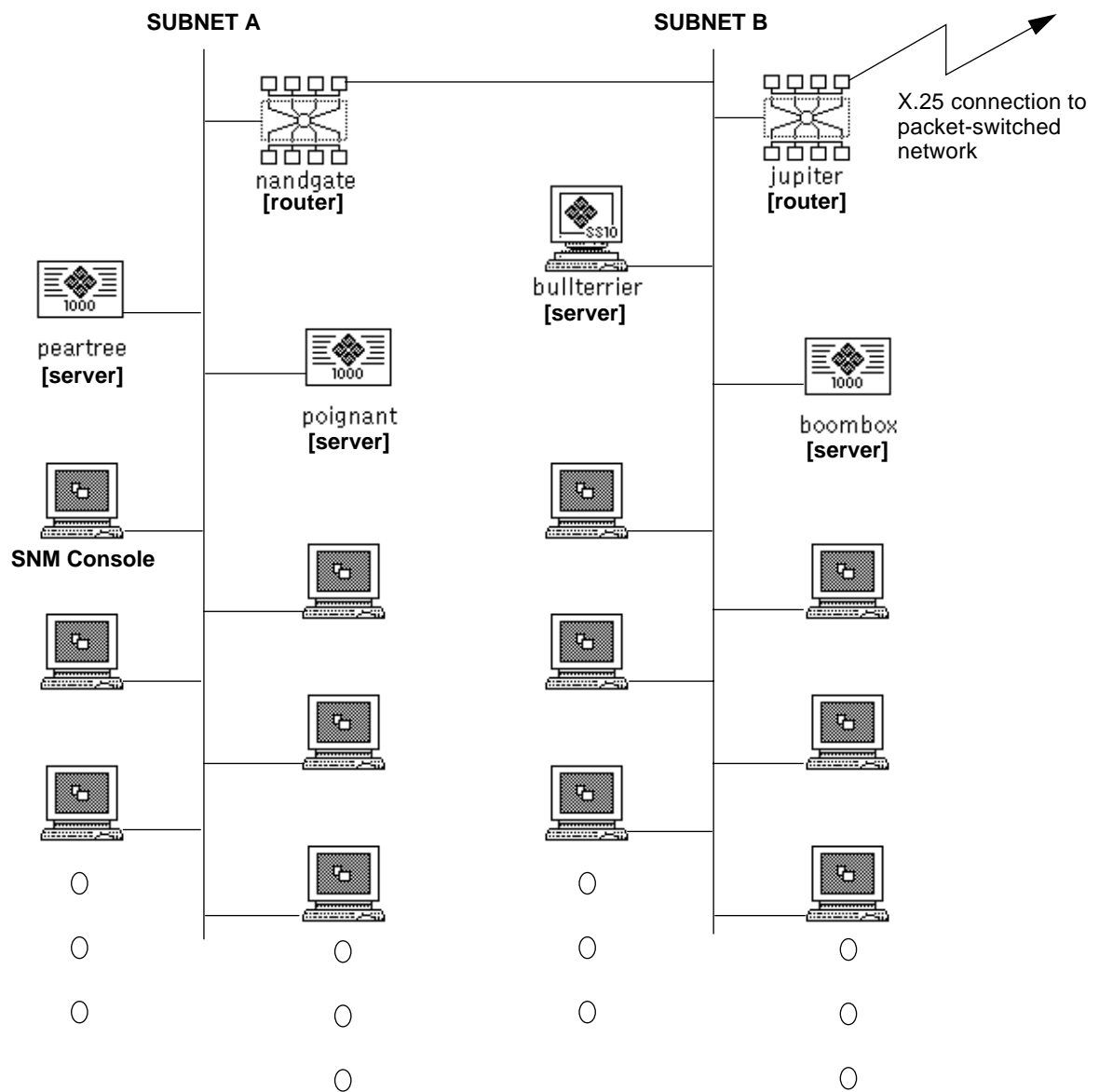


Figure 2-1 Example of a LAN Configuration

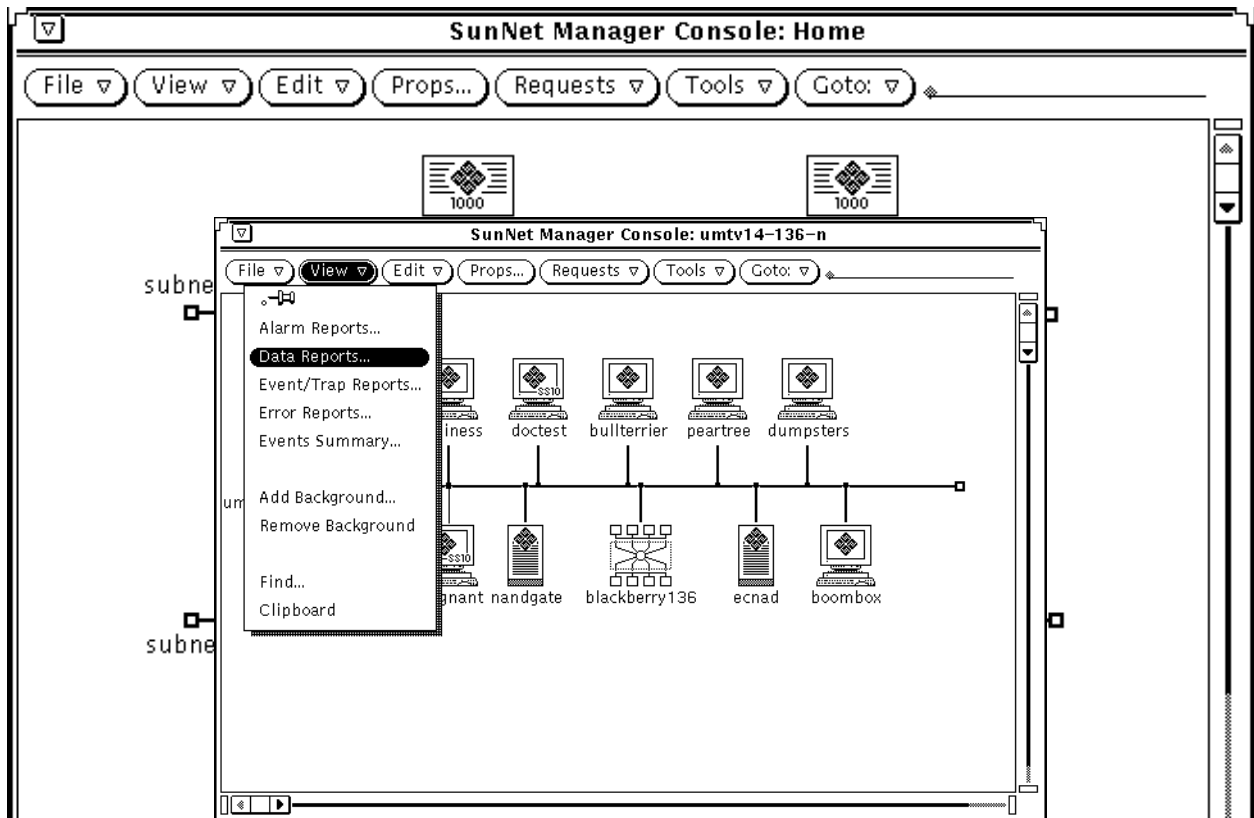


Figure 2-2 Critical Nodes in Home View

The cloud glyphs in Figure 2-3 represent the separate views into the LAN and its subnetworks. Figure 2-4 shows the elements within these clouds. The same element can occur within multiple views. A software server, such as poignant, can occur both in the Servers view and in the Net_B view, which shows all nodes in a subnetwork.

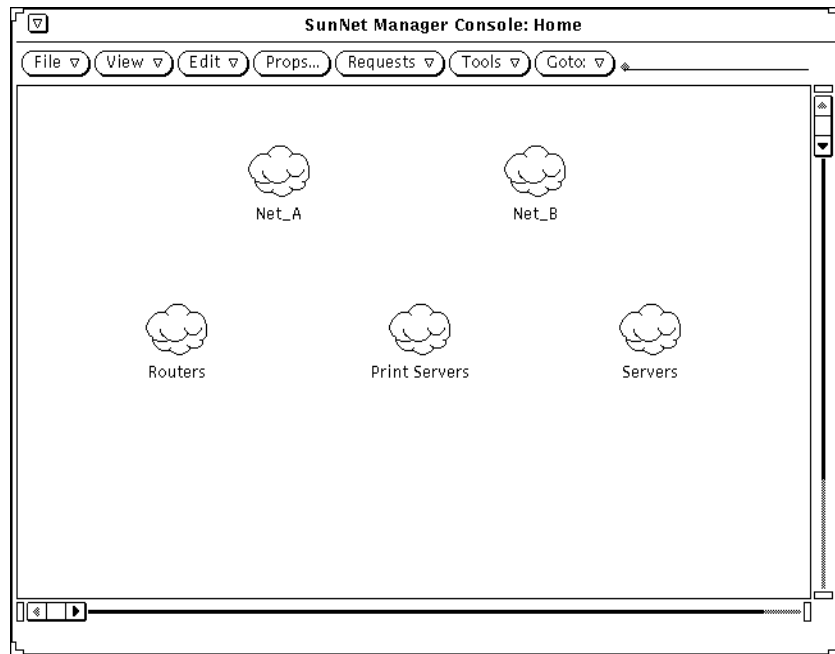


Figure 2-3 SunNet Manager Console Views

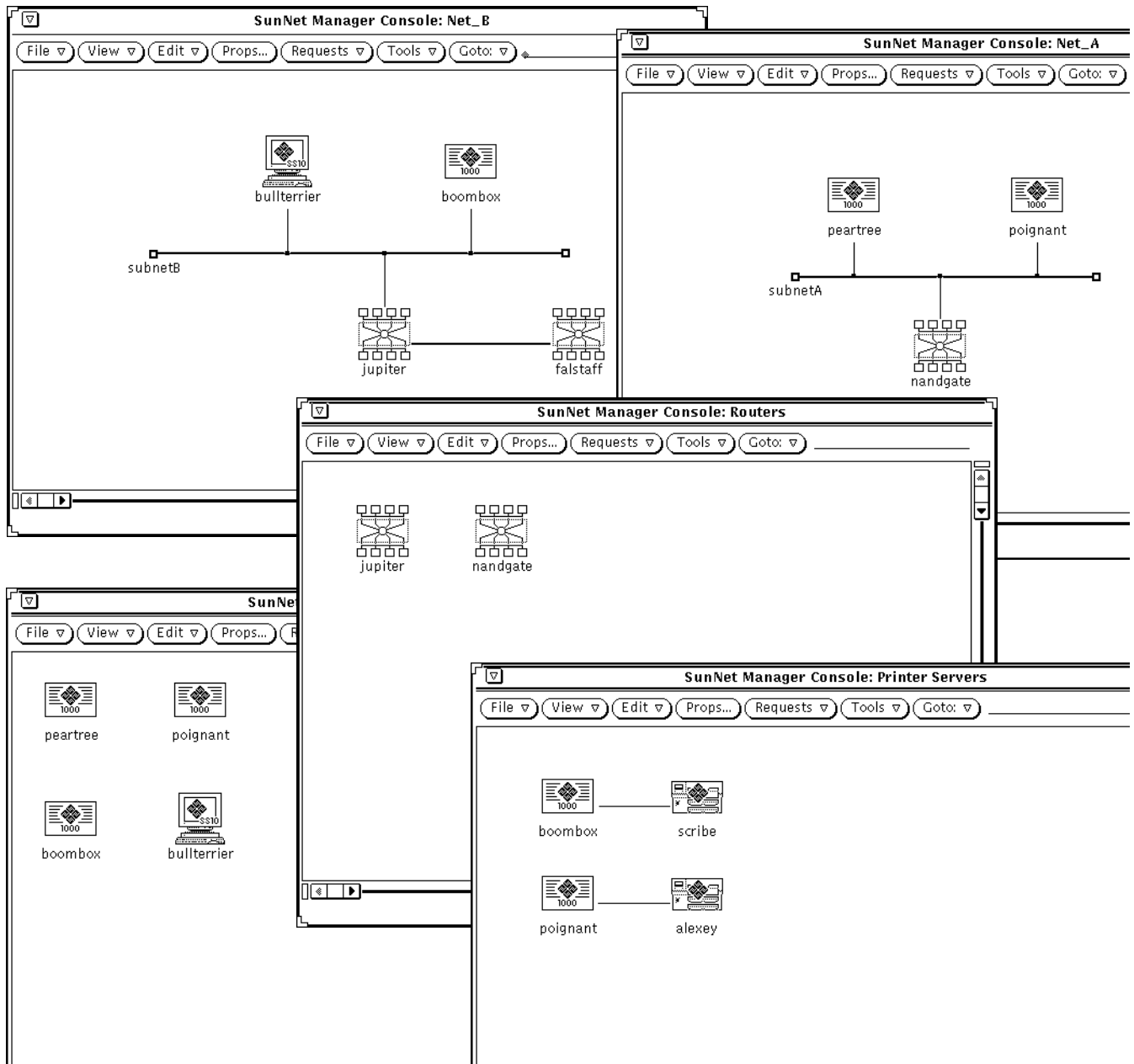


Figure 2-4 Elements Within Console Views

3. What type of information do we need to use this product effectively?

SunNet Manager provides three types of information:

a. Event notification

You want to be notified of critical events. The event request mechanism allows you to define conditions that generate an event notification. After it is specified, the event request is sent to the agent. The agent generates an event whenever the specified condition becomes true.

If one of your critical nodes becomes unavailable to users, you will want to know this immediately. This means you must specify frequent polling intervals, such as every five minutes, in the appropriate event requests.

If you want to know the status of a certain device, for example, whether a router is currently down and whether it has been down at any time since you last cleared an event, you want to choose Color By Priority as the method for notifying you. You can then take advantage of the “decay” feature. When the specified event occurs, its glyph changes to the priority color you have selected, then changes to blue, or the color you select, if the condition (e.g., unavailability) is no longer true.

Starting with this release of SunNet Manager, you are not limited to yellow, orange, red, and blue as the colors representing low, medium, high priority, and the decay state. You can customize colors to create ones that you prefer. See Chapter 5, “Specifying Event Requests” or Chapter 4, “Requesting Data” for more information.

Also starting with the current release, you can put a glyph in pending state to avoid logging repeated events or traps against the device it represents. This is useful when a device has been down for an extended time. When pending state is turned off, the current status of the device is displayed. See Chapter 5, “Specifying Event Requests,” for more information.

In addition to events generated by agents in response to event requests, there are also unsolicited events — called “traps” — generated by Simple Network Management Protocol (SNMP) devices, such as routers. For information on how to set up SunNet Manager to receive traps, refer to Chapter 8, “Managing SNMP Devices.”

b. Data reporting

Over a period of time, statistical information can help you compare the performance of your critical nodes, such as servers and gateways. Data reporting provides this information in addition to statistics on network traffic levels at various times. Collecting data on resource utilization rates can help you determine whether particular network resources are being overburdened.

In most cases, you will probably not want continual polling for data, as this can add an additional burden to the network. If you periodically activate data-gathering periodically, you can develop a historical record to help you spot trends and determine “normal” load levels.

c. Topology information

You also want information on changes to the network’s topology, such as the addition of new devices and connections. IP Discover tool’s Monitor function searches the network to provide this information. You might want to limit these to weekly searches, since network configuration typically changes slowly.

Starting with this release, you can use IPX Discover to search your network for Novell Netware IPX nodes. See Chapter 23, “IPX Discover,” for more information.

4. What agents can provide the type of information we want?

After you know the type of event or data reports you need, you can determine which agents are required.

Table 2-1 provides a summary of the agents shipped with the product. In some cases, agents return information similar to that provided by certain UNIX commands; this is indicated in the table.

Table 2-1 Summary of Shipped Agents and Proxies

Name	Description	Related UNIX Command	Type
cpustat	Gather CPU statistics for multiprocessor systems		Agent
diskinfo	Reports disk usage information	df	Agent
etherif	Ethernet interface statistics for SunOS 4.x clients	—	SunOS 4.x agent
etherif2	Ethernet interface statistics for Solaris 2.x clients	—	Solaris 2.x agent
hostif	Monitors interfaces that send IP packets	netstat -i	Agent (see note)

Table 2-1 Summary of Shipped Agents and Proxies

Name	Description	Related UNIX Command	Type
hostmem	Memory utilization information for SunOS 4.x clients	netstat -m	SunOS 4.x agent
hostmem2	Memory utilization information for Solaris 2.x clients	netstat -m	Solaris 2.x agent
hostperf	Host system performance data	rup and perfmeter	Proxy agent
iostat	Input/output statistics for SunOS 4.x clients	iostat	SunOS 4.x agent
iostat2	Input/output statistics for Solaris 2.x clients	iostat	Solaris 2.x agent
ippath	IP packet trace information	—	Proxy agent
iproutes	IP route table and statistics	netstat -r	Agent
layers	Protocol layer statistics for SunOS 4.x clients	netstat -rs netstat -s	SunOS 4.x agent
layers2	Protocol layer statistics for Solaris 2.x clients	netstat -rs, netstat -s	Solaris 2.x agent
lpstat	Printer status	lpq and lpstat	Proxy agent
ping	IP connectivity information	ping	Proxy agent
rpcnfs	Remote Procedure Call and Network File System statistics	nfsstat	Agent
snmp	Information from MIB I-compliant SNMP devices.	—	Proxy agent
snmpv2	For managing SNMP Version 1 and Version 2 devices	—	Proxy agent
snmp-mibII	Information from MIB II-compliant SNMP devices.	—	Proxy agent
sun-snmp	MIB I-compliant and Sun-specific information from Sun workstations.	—	Proxy agent
sync	Synchronous serial lines monitoring	syncstat	Agent
traffic	Ethernet traffic analyzer	—	Proxy Agent

After you have installed SunNet Manager, you can get specific information about each agent by consulting the man page for na.<agent-name>. For example:

```
hostname% man na.ping
```

To have access to the man pages, make sure you have set your MANPATH environment variable, as described in your installation manual.

In the case of the `na.snmp` proxy agent, there are three schema files shipped with SunNet Manager that can be used with it (`snmp`, `snmp-mibII`, and `sun-snmp`). You can get information on the `snmp` and `snmp-mibII` schema files in the man pages for `<agent-name>.schema`. For example:

```
hostname% man snmp-mibII.schema
```

The `sun-snmp.schema` file supports the features of the Sun enterprise-specific SNMP agent, `snmpd`, for Sun workstations and servers. To access information on this agent, enter the following command:

```
hostname% man snmpd
```

An efficient arrangement for proxy agents is to distribute them to collect information within separate domains or subnetworks. This reduces network traffic between the management station and managed devices.

For example, to find out if a router is down in a remote subnet, the management station can send an event request to a `ping` proxy agent in the target subnet. The proxy agent responds if it detects that a router is not available. Regular polling of routers is limited to the subnet where the proxy system is located.

For information on installing the SunNet Manager agents, refer to your installation manual.

Creating and Modifying the Management Database



This chapter discusses the following topics:

- Starting and quitting the Console
- Creating the initial runtime database
- Modifying the database when elements and agents are added, deleted, or changed.

Use the Console to create a graphic representation of elements in a glyph. This graphic representation of your network is reflected in the *runtime management database*, which holds all your network elements, possible element types, and predefined requests.

Note – All command line examples in this Guide use the C-shell environment. If you use the Bourne or Korn shells, refer to the `sh(1)` or `ksh(1)` man pages, respectively, for compatibility information.

3.1 Adding SunNet Manager to Your PATH Variable

Use the command line to start SunNet Manager. To do this, you could type the full path name.

For example, if SunNet Manager was installed on a Solaris 2.x machine in the default location, you could enter:

```
host% /opt/SUNWconn/bin/snm &
```

However, to avoid typing the complete path each time you start the Console, set the PATH environment variable in your `.cshrc` file (or `.profile` file if you use a Bourne or Korn shell) to point to the location of the executable files (the SunNet Manager Console, and tools). For a C-shell, enter the following in your `.cshrc` file:

For Solaris 1.x:

```
setenv PATH ${PATH}:/usr/snm/bin
```

For Solaris 2.x:

```
setenv PATH ${PATH}:/opt/SUNWconn/bin
```

For a Bourne or Korn shell, consult the examples in your installation guide. Examples in the rest of this chapter assume that your PATH environment variable points to the appropriate location.

Note – If you installed SunNet Manager in a location other than the default (`/usr/snm` on Solaris 1.x machines, `/opt/SUNWconn/snm` on Solaris 2.x), you must set the environment variable `SNMHOME` to the installation directory before invoking the command to start SunNet Manager. See your installation guide for instructions for setting this variable.

3.2 Starting the Console

If no management database exists when you start the Console — which is the case for first-time users—you receive a Quick Start window show in Figure 3-1.

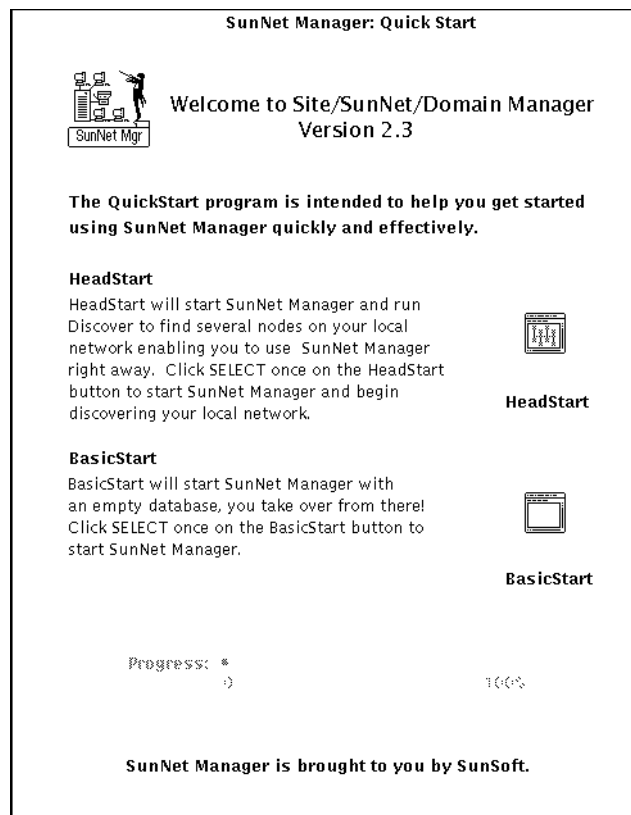


Figure 3-1 Quick Start Window

If a database exists, you go directly to the SunNet Manager Console window shown in Figure 3-2.

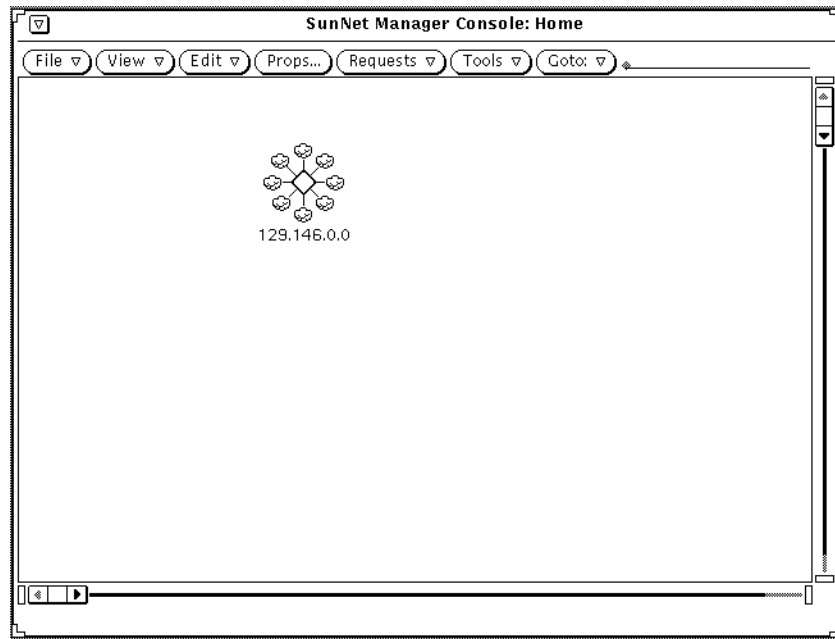


Figure 3-2 Console Window

Note – You can use the `-q` option to suppress the Quick Start window.

3.2.1 Startup Commands

- To start the Console when there is no database or to restart the Console with the runtime management database from the last Console session (assuming no intervening reboot), enter

```
host% snm &
```

Appending an ampersand (&) to the command line puts the command in the background.

- To start the Console and initialize (in effect clearing) the runtime database, enter the command below. You receive the Quick Start window.

```
host% snm -i &
```



Caution – Use the `-i` option with care. If you have created a runtime management database in a prior Console session, the `-i` option deletes this database.

- You can use the `-i` option in conjunction with the file name of an ASCII database file. For example:

```
host% snm -i /home/alex/snm/osaka.dbase &
```

This command deletes a runtime database, and loads the specified ASCII database file into the runtime database. The Quick Start window does not appear.

- A variation of the preceding command example is to specify an ASCII database, but not the `-i` option. The specified ASCII database file is merged into the runtime database. For example:

```
host% snm /home/alex/snm/osaka.dbase &
```

- The `snm` command has two additional command-line options, `v1` and `v2`. Use the `v1` option if you are loading database files used with SNM 1.x. Use the `v2` option only if you are loading multiple database files, one or more, **but not all**, of which are for SunNet Manager version 1.x. The `v2` option allows you to distinguish those files that are not from version 1.x. For example:

```
host% snm -v1 /home/alex/snm/west.dbase -v2 /home/alex/snm/osaka.dbase
```

3.2.2 Managing Duplicate Databases

Starting with version 2.3, if a database you specify has elements in common with a runtime database already present, you can either abort the database you are loading, ignore the duplicates and proceed with the load, or replace the old database with the new one from the ASCII database. See Chapter 18, “Management Database,” for more information.

3.2.3 Disk Space Requirements

A minimum of 10-15 Mbytes of disk space should be available in the directory that stores the runtime database. You can check the available disk space with the following command:

- `df -a` (if you're running Solaris 1.x)
- `df -k` (if you're running Solaris 2.x)

You specify the database location during installation of SunNet Manager or by using the environment variable `SNMDBDIR`. If you have not specified `SNMDBDIR`, then the following directory is assumed:

- `/var/adm/snm` for Solaris 1.1.1
- `/var/opt/SUNWconn/snm` for Solaris 2.4

The disk space required may vary according to the size of the network being discovered. For example, a database that contains 647 components, 9 buses, and 11 views takes approximately 3.2 Mbytes of disk space. If your runtime database grows very large, your file system can run out of space. This can be a problem if the `/var` directory is in the root partition, which generally does not have a large amount of free disk space. In this case, you can change the directory specified by the `SNMDBDIR` environment variable to point to another partition with more free disk space.

Note – When the database directory is changed, the files under the current `SNMDBDIR` should be moved to the new location to ensure smooth operation.

The new directory should be created with mode 777 so that the database files can be written to the directory. When you run the IP Discover Tool, both you (as user) and root must be able to write to this directory—therefore, you should locate the database on a local file system rather than an NFS-mounted file system. (NFS servers usually do not allow root users on remote machines to have write permission.)

3.2.4 Quick Start Window

After invoking SunNet Manager with the `-i` option or with no command-line arguments when there is no database present, you receive a Quick Start window.

You can suppress the Quick Start window by using the `-q` option. This option works when you invoke `snm` with no other options or with the `-i` option.

The Quick Start window has the HeadStart and BasicStart options. The Progress bar is dimmed. If you are a first-time user, HeadStart is recommended.

Clicking SELECT on the buttons has the following effects:

- The HeadStart button launches the SunNet Manager Console and discovers a few nodes (up to a maximum of ten) on the local subnetwork, so that you can begin using SunNet Manager immediately.
- The BasicStart button skips the discovery of local nodes and directly invokes the SunNet Manager Console window.

3.2.5 Console Window Characteristics

You can configure a number of Console window characteristics. For example, you can change text that appears in the title bar. For more information on changes you can make, see “Part 2: Reference.”

You initiate most SunNet Manager activities through pull-down menus accessed in the abbreviated menu buttons in the Console’s control area. The effect of a menu selection differs depending on whether or not you select a glyph in the Console window. In general, if you select a glyph, your menu choice affects only the highlighted element. If no glyph is selected, your menu choice either affects all elements or has no consequences for any single element.

3.3 Using IP Discover and IPX Discover

This section discusses the discover function of the IP Discover and IPX Discover tools. You use these functions normally when you first bring up the Console. More details on these tools is in Chapter 22, “IP Discover” and Chapter 23, “IPX Discover.”

3.3.1 Using IP Discover

The IP Discover tool, invoked from the Tools menu in the Console window or from a command line, finds networks, subnetworks, routers, and hosts. By default, IP Discover limits search to the local subnetwork to which your SunNet Manager machine’s Console is attached.

The IP Discover tool has two functions:

- *IP Discover*, which finds hosts, routers, networks, subnetworks, and Simple Network Management Protocol (SNMP) devices reachable from the Console machine. IP Discover stores a record in the runtime database for each element it finds.

- *Monitor*, which compares elements stored in the runtime database with the elements found at a specified interval or specified time. If new elements are detected, the monitor function stores elements in a holding area view (which you can name) and records these elements in a log file.

Both the discover and monitor functions have configuration windows, available through the Configuration Options button, that allow you to fine-tune the depth and breadth of the activities.

3.3.1.1 Invoking IP Discover

Use one of the following two methods to invoke the IP Discover tool:

1. Press **MENU** on the **Tools** button in the **Console** window.
2. Press **MENU** on **Discover**.
3. Release **MENU** over **IP Discover**.
4. You see the window shown in **Figure 3-3**.

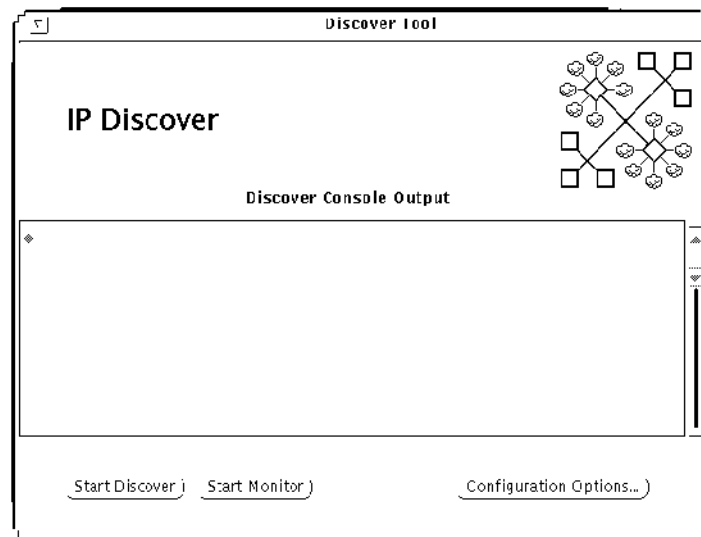


Figure 3-3 IP Discover Home Screen

5. Click SELECT on Start Discover to start the IP Discover function.

The Start IP Discover button changes to Stop IP Discover. The IP Discover function continues to run, within the boundaries you have configured in your local subnetwork until it finds all elements within the subnetwork

6. Click SELECT on Stop Discover.

From the SunOS command line, if you have the path to the SunNet Manager executables in your PATH variable, invoke `snm_discover`. For example, to run IP Discover and invoke the user interface, enter:

```
hostname# snm_discover -T
```

The path to `snm_discover` is:

- `/usr/snm/bin` on SunOS 4.x
- `/opt/SUNWconn/snm/bin` on Solaris 2.x

You can start IP Discover directly, bypassing the initial Home window if you use the `snm_discover` command without the `-T` option.

3.3.1.2 *IP Discover Views Beyond the Local Subnetwork*

IP Discover reports progress in the IP Discover Console Output window and adds elements to the current view in the SunNet Manager Console window. Simultaneously, it adds these elements to the runtime database.

Running IP Discover with the Default option will find network elements in the subnetwork to which the Console system is attached. If you enable IP Discover to reach beyond your local subnetwork, it constructs a hierarchy of views. See Chapter 22, “IP Discover,” for more information. The Home view is the top level in the hierarchy of your management domain views.

1. Double-click the SELECT button on each network/subnetwork glyph to display the elements, usually hosts and routers, contained in each of these views.

The console output section of the IP Discover window displays information about the program’s progress.

You can display logical groupings of elements, such as routers in your management domain, or all the SNMP devices in a particular subnet.

2. Click SELECT on the Configuration Options button to display the Properties sheet.

Figure 3-4 shows the Properties sheet. For a description of fields, see Chapter 22, “IP Discover.”

The screenshot shows a dialog box titled "Discover Configuration" with the following fields and options:

- Category: Discover
- Net Name/Number: mpk16-185-n
- Netmask: 0xfffff00
- Viewname: Home
- Maximum Hops: 0
- ICMP Retries: 2
- ICMP Timeout: 1
- Add To All Views: Yes No
- Add Object Connections: Yes No
- Add Object Coordinates: Yes No
- SNMP Community: public
- SNMP Retries: 3
- SNMP Timeout: 3
- Default Proxy: localhost
- Search Method: DEFAULT ARP PING
- Gateway Filename:
- Verbose Mode: Yes No
- Objects To Discover:
 - All Objects
 - SNMP hosts
 - SNMP devices
 - Routers
 - Networks/Subnets

Buttons: Apply, Reset

Figure 3-4 IP Discover Properties Sheet

3.4 Using IPX Discover

Starting with version 2.3 of SunNet Manager, you can use IPX Discover to discover IPX Netware nodes and networks. See “Part 2: Reference” for detailed information on IPX Discover functionality.

3.5 *Netware Management System Export/Import Agent*

IPX Discover discovers and models IPX networks by communicating with a topology export/import agent (NXIS) located on the Novell Management platform (Managewise). NXIS exports topology data discovered by Managewise; Managewise need not be running during this process.

IPX Discover has scheduling capability and can be configured to periodically poll the NXIS agent for changes to the network.

Refer to the IPX Discover man page for command syntax.

3.6 *Invoking IPX Discover*

To invoke IPX Discover, follow the steps below

1. Click **SELECT** on the **Tools Menu**►**Discover**►**IPX Discover** to see the screen in **Figure 3-5**.

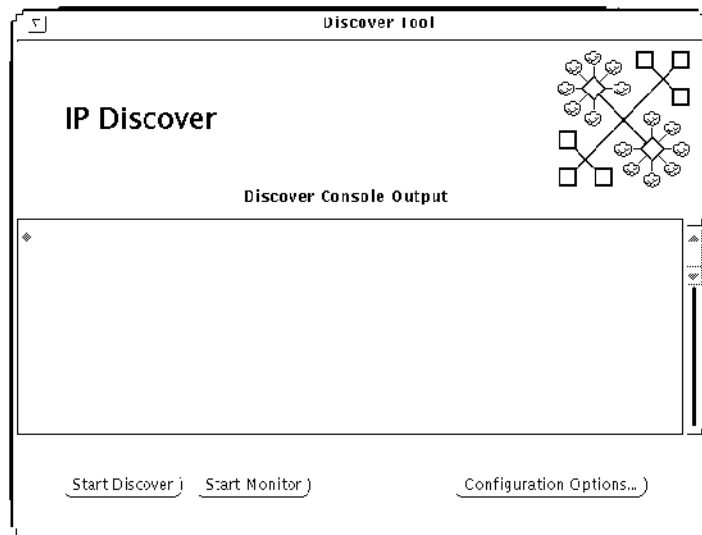


Figure 3-5 IPX Discover Home Screen

2. Click **SELECT** on Configuration to see the Properties sheet in Figure 3-6.

IPX Discover Tool Configuration

Add Object Connections: Yes No

Export Service Agent

Server Name: _____

Enter Password: _____

Confirm Password: _____

IP Address: _____

Schedule Discover:

Date Format:

Start Date: _____

Stop Date: _____

Start Time: _____

Stop Time: _____

Period Time: _____

Apply Reset

Figure 3-6 IPX Discover Properties Sheet

3.6.1 IPX Discover Options

From this point, you can use IPX Discover any of the following four ways:

- A. Perform a load to export a database from the Export Services Agent without using the scheduler.
- B. Perform a load using the scheduler without specifying time period to export the database.

C. Perform an update using the scheduler and specifying how often to report changes that have occurred on the Novell network.

D. Perform multiple loads and updates from multiple Export Services Agents.

Steps for each of these options are provided below.

A. Load without using the scheduler:

1. On the IPX Discover Properties sheet, enter the Server Name of the Novell export agent.
2. Enter the password for the Novell export agent.
3. Enter the IP address of the console on which the export agent resides.
4. Click SELECT on Add►Apply.
5. On the IPX Discover Home screen, click SELECT on Discover.

B. Load using the scheduler without specifying period time:

1. On the IPX Discover Properties sheet, enter the Server Name of the Novell export agent.
2. Enter the password for the Novell export agent.
3. Enter the IP address of the console on which the export agent resides.
4. Click SELECT on Add.
5. Click SELECT on Schedule Discover.
6. Enter Start and Stop Dates in the appropriate date format.
7. Click SELECT on Start Time and Stop Time, and choose the desired times.
8. Click SELECT on am or pm button.
9. Click SELECT on Add►Apply.
10. On the IPX Discover Home screen, click SELECT on Discover.

C. Update using the scheduler and specifying period time:

1. On the IPX Discover Properties sheet, enter the Server Name of the Novell export agent.

2. Enter the password for the Novell export agent.
3. Enter the IP address of the console on which the export agent resides.
4. Click SELECT on Add.
5. Click SELECT on Schedule Discover.
6. Enter Start and Stop Dates in the appropriate date format.
7. Click SELECT on Start Time and Stop Time, and click on the desired times.
8. Click SELECT on am or pm button.
9. Click SELECT on Period Time button, and click on the desired time.
10. Click SELECT on am or pm button.
11. Click SELECT on Add►Apply.
12. On the IPX Discover Home screen, click SELECT on Discover.

D. Perform multiple loads and updates with multiple export agents:

1. On the IPX Discover Properties sheet, enter the Server Name of the first Novell export agent.
2. Enter the password for the Novell export agent.
3. Enter the IP address of the console on which the export agent resides.
4. Click SELECT on Add.
5. For each additional export agent, follow steps 1 - 3.
6. Click SELECT on Schedule Discover.
7. Enter Start and Stop Dates in the appropriate date format.
8. Click SELECT on Start Time and Stop Time, and click on the desired times.
9. Click SELECT on am or pm button.
10. Click SELECT on Period Time button, and click on the desired time.
11. Click SELECT on am or pm button.

12. Click **SELECT** on **Add►Apply**.
13. On the **IPX Discover Home** screen, click **SELECT** on **Discover**.

3.6.2 *Change, Remove, and Reset Options*

To change the configuration of an IPX Discover, enter new information in the appropriate fields and click **SELECT** on **Change►Apply►Start Discover**.

To remove a configuration from the Export Agent Services list, use the mouse to highlight your choice, then click **SELECT** on the **Remove** button.

To reset your configuration if you *have not* clicked on the **Add** button, click **SELECT** on the **Reset** button. All fields become blank. If you *have* clicked on the **Add** button, click **SELECT** on the **Change** button.

3.7 *Traversing the View Hierarchy*

Whenever you start the Console, the Home view is displayed. Elements that are views can contain other elements, including other views. Other types of elements, such as components, can also contain other elements. The Console keeps track of up to 16 different views or elements as they are displayed. You can traverse through the Console view hierarchy in several ways.

1. **If the element is a view, double-click SELECT over the glyph that represents the element to display the elements contained in the view.**
2. **To return to any of the views that have already been displayed, press MENU on the Goto button and release MENU over the name of the view you want to display, as indicated in Figure 3-7.**

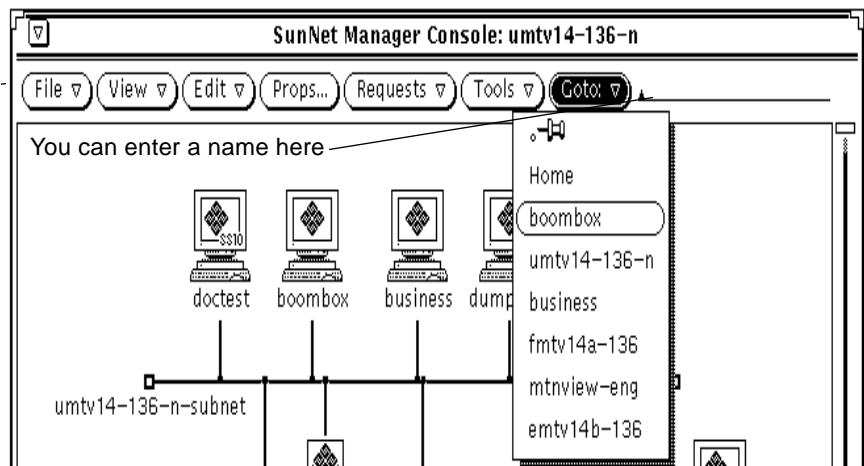


Figure 3-7 Selecting a View Name in the Goto Menu

3. To return to the last view that was displayed, click **SELECT** on the **Goto** button. The default is the first item on the menu, **Home**, in this case

3.7.0.1 Selecting an Element or View by Name

- If you know the name of an element in a view, you can use the Find option in the View menu to display the view(s) that contain the particular element. See the Section on “Finding Elements” later in this Chapter.
- If you know the name of the view you want to see, type in the view name on the line next to the Goto button and press Return. If the view name you specify is not part of the runtime database, the Console displays a message “View name <view_name> does not exist” in the Console window footer area and leaves you in the current view.

3.7.1 Returning to the Home View

1. Press **MENU** on the **Goto** button; choose the **Home** option.
 - a. Or, type “Home” in the Goto line and click **SELECT** on the **Goto** button.

If you often find it necessary to return to the Home view, you could have the Home view displayed whenever you click SELECT on the Goto button. This change is made in the Console Properties window Miscellaneous category. Refer to “Part 2: Reference” for more information.

3.8 Creating Elements Using the Editor

This section describes the Console’s graphical editor function, accessible through the Edit>Create button. This function limits you to creating elements one-at-a-time. Use IP Discover as a more convenient way to create elements in large networks.

You can create new elements of different types to represent particular views, networks, devices, and other components in your network. Elements are represented by glyphs, a type of icon, which appear in one or more views. To create new elements follow the steps below:

1. In the Console’s control area, press MENU on the Edit button.
2. Release MENU over the Create item. (Alternatively, you can press MENU over an empty space in the view to obtain a floating popup menu. You receive the window shown in Figure 3-8.

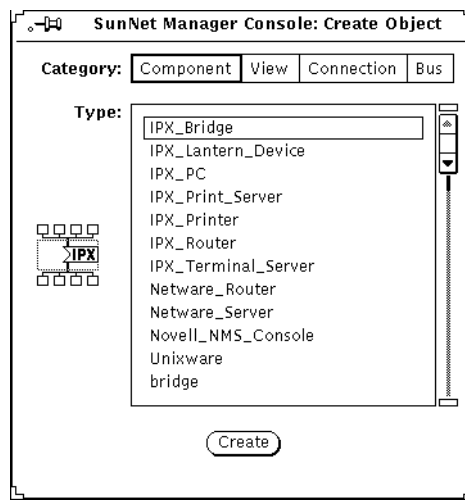


Figure 3-8 Create Object Window

3. Click **SELECT** on your choices for Category and Type.

4. Click **SELECT** on Create. If you select a Component as a category, you receive the window shown in Figure 3-9.

Screenshot of the SunNet Manager Console: New component.IPX_Bridge window. The window contains the following fields and controls:

- Name: _____
- IP Address1: _____
- IP Address2: _____
- Contact: _____
- User: _____
- Location: _____
- Description: _____

Below the fields is a list of categories with checkboxes:

<input type="checkbox"/>	HOST-RESOURCI	HOST-RESOURCES-MIB agent
<input type="checkbox"/>	Host-Resources	Host Resources
<input type="checkbox"/>	RFC1213-MIB	RFC1213-MIB agent
<input type="checkbox"/>	RFC1229-MIB	RFC1229-MIB agent
<input type="checkbox"/>	RFC1230-MIB	RFC1230-MIB agent
<input type="checkbox"/>	RFC1231-MIB	RFC1231-MIB agent
<input type="checkbox"/>	RFC1232-MIB	RFC1232-MIB agent
<input type="checkbox"/>	RFC1233-MIB	RFC1233-MIB agent
<input type="checkbox"/>	RFC1253-MIB	RFC1253-MIB agent
<input type="checkbox"/>	RFC1271-MIB	RFC1271-MIB agent

At the bottom, there are three sliders for Red, Green, and Blue, and buttons for Apply, Reset, Alias..., and Create.

Figure 3-9 Object Properties Window (Component)

The top portion of the window is the element data.

1. Complete the Name field using a valid IP address or name. This field is required.

If you enter a name, it should be in a local operating-system configuration file (such as `/etc/hosts`) or in a directory service map or table. SunNet Manager does not check for the correctness of an IP address or name until you send a request to that element.

Other fields are optional; you can use them to record information about the element for your reference.

The middle portion of the new-element window is the list of agent schema files that the Console knows about. This and the bottom portion are the same for all categories of elements as shown in Figure 3-9.

In the middle portion, click SELECT on the box to the left of a listed schema to toggle a check mark—this specifies that the element can be managed with this schema. Check all schemas that apply to the element. The line to the right of some schema names indicates that a proxy system can be defined. If you do not define a proxy system, the default proxy system will be the system on which you are running the Console.



Note – Checking a schema does not make the element manageable from the Console; the appropriate agent software must be installed and running on the target system. See your installation guide for instructions on running the `getagents` script to install agents. The exceptions to this are the `hostperf` and `ping` proxy agents. On most Sun and those of many other vendors on a TCP/IP network, you can check off the `hostperf` and `ping` agents and use the Console machine (`localhost`) as the proxy.

The bottom portion of the Properties window allows you to specify a color for the glyph associated with the element.

When you finish entering data and making selections in the new-element window, click SELECT on Apply. A glyph for the element you just defined appears in the current view with the name you specified.

Figure 3-10 shows the data portion of the Object Properties windows for the remaining element categories: view, connection, and bus.

SunNet Manager Console: New view.building

Name: _____

Description: _____

Glyph State: Inherited From Children

Inherited From Children

Not Inherited

Data portion of Object Properties Window for a new view

SunNet Manager Console: New connection.link

Name: _____

Object1: _____

Object2: _____

Description: _____

Data portion of Object Properties window for a new connection

SunNet Manager Console: New bus.ethernet

Name: _____

IP Network Number: _____

Description: _____

Glyph State: Inherited From Children

Default Proxy: _____

Inherited From Children

Not Inherited





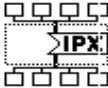

Data portion of Object Properties Window for a new bus

Figure 3-10 Top Portions of Object Properties Windows

3.8.0.1 Glyphs for Multiple Element Types

The glyphs in Table 3-1 are used for more than one type of element.

Table 3-1 Glyphs Used for Multiple Element Types

Glyph	Element Category	Element Type
	Component	ss330 ss370 sun-deskside
	Component	sun-server sun470
	Connection	link rs232
	Component	IPX-Latern Device IPX-PC Unixware
	Component	IPX-Bridge IPX-Router Netware Router
	Component	IPX Print Server IPX Printer

3.8.1 Useful Information About Creating Elements

SunNet Manager Console: New component.IPX_Bridge

Name: _____

IP Address1: _____

IP Address2: _____

Contact: _____

User: _____

Location: _____

Description: _____

<input type="checkbox"/>	HOST-RESOURCI	HOST-RESOURCES-MIB agent
<input type="checkbox"/>	Host-Resources	Host Resources
<input type="checkbox"/>	RFC1213-MIB	RFC1213-MIB agent
<input type="checkbox"/>	RFC1229-MIB	RFC1229-MIB agent
<input type="checkbox"/>	RFC1230-MIB	RFC1230-MIB agent
<input type="checkbox"/>	RFC1231-MIB	RFC1231-MIB agent
<input type="checkbox"/>	RFC1232-MIB	RFC1232-MIB agent
<input type="checkbox"/>	RFC1233-MIB	RFC1233-MIB agent
<input type="checkbox"/>	RFC1253-MIB	RFC1253-MIB agent
<input type="checkbox"/>	RFC1271-MIB	RFC1271-MIB agent

Red: 0 _____

Green: 0 _____

Blue: 0 _____

Apply Reset Alias...

Create

Figure 3-11 Object Properties Window

3.8.1.1 IP Address

The IP Address field in the Properties window for the element is not used to manage the device—it is for your information only. See Figure 3-11 for location of the IP Address field.

3.8.1.2 Proxy Agent System Name

The blank line next to some agent schema names allows you to specify the name of the system on which the proxy agent resides. (For example, in Figure 3-11, blank lines exist next to each schema file on the left side of the window.) This system name is for a default proxy system. You can specify a different proxy system name for data reports and when specifying events. If

you do not specify a proxy system in the Properties window or in the report request, the system on which the Console is running is assumed to be where the proxy agent resides.

3.8.1.3 Creating Elements Within Elements

You can create an element that logically contains other elements. For example, if managing all the file servers in your management domain is important, you might want to create a view called File Servers. To do so, perform the following steps:

- 1. Create an element of category View, as described previously.**
- 2. Double-click SELECT on the new glyph.**
- 3. In the empty view, create the element instance for each device. If the element instance definition already exists, copy the glyph into the view. (See the Section on “Copying Elements” later in this chapter.)**

3.8.1.4 Changing Label Fonts

You can change the font used for the labels under element glyphs. This is done in the Console Properties window. (If you change the font, you must restart the Console for the change to take effect.) See the description of the Icon Font setting in “Part 2: Reference.”

3.9 Creating Aliases

SunNet Manager has an alias feature that allows you assign multiple names to an element. This feature is useful with machines that have multiple network interfaces (usually routers). You can use an alias in the same way that you use element names. That is, you can search on an alias or send requests that specify an alias.

If you use unique names for each interface, the Discover tool fills in aliases automatically when it creates a router. The first name it finds is the one filled in as the name of the element and the names of subsequently-found network interfaces as aliases.

To add one or more aliases for an element, do the following:

- 1. Press MENU over the glyph to which you want to add aliases.**
You can also click SELECT on the glyph to highlight it, then press MENU over the glyph. You receive the Properties window for that element.
- 2. In the Properties window, click SELECT on the Alias button.**
You receive the window shown in Figure 3-12.

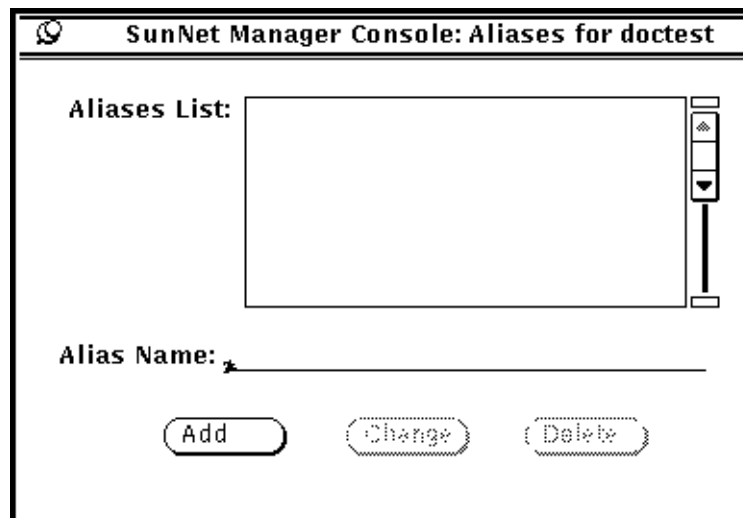


Figure 3-12 Alias Window

- 3. Enter a name and click SELECT on Add.**
The name you entered appears in the Aliases List.
- To change or delete an alias:
- 1. Click SELECT on the alias in the Aliases List**
 - 2. Click SELECT on Change or Delete.**

The same rules apply to aliases as to element names, as specified in the subsection on “Creating Elements” later in this chapter. You can use an IP address or a name.

The Properties window for a router has two IP address fields as shown in Figure 3-13.

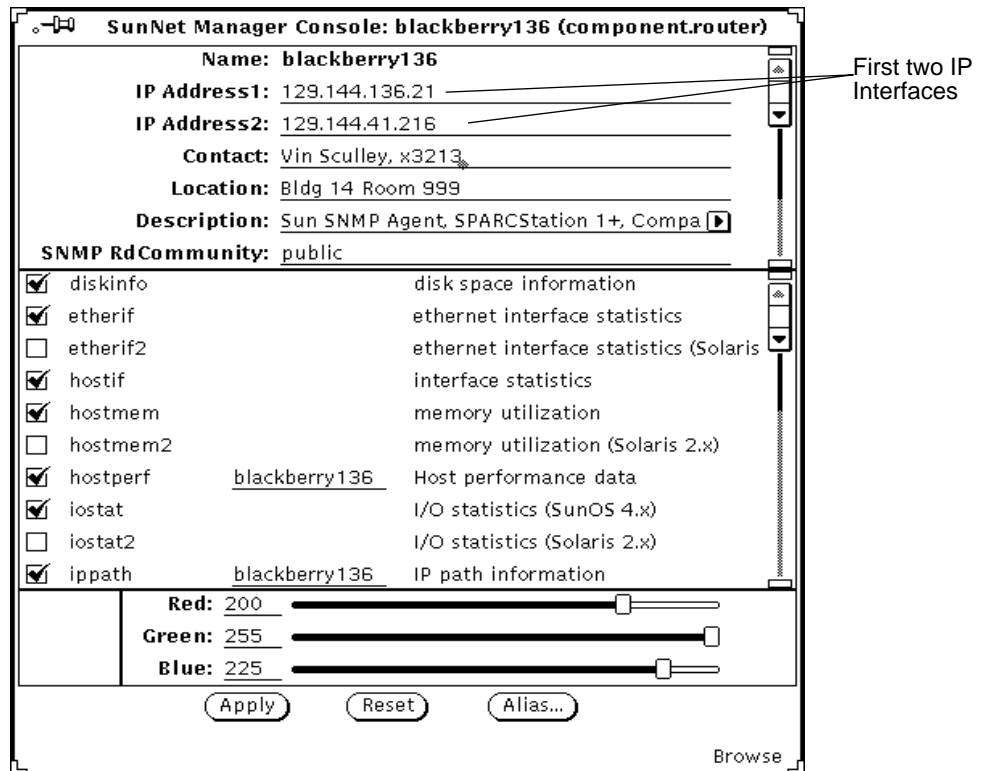


Figure 3-13 Properties Window for Router

You are not, however, limited to managing the two interfaces identified in the IP Address fields. The alias feature allows you to add a name for each interface on a machine. Note that the IP Discover tool does this automatically.

3.10 Finding Elements

The Find option of the View menu displays the view(s) in which a specified element is located.

1. In the Console window, invoke **View>Find**.

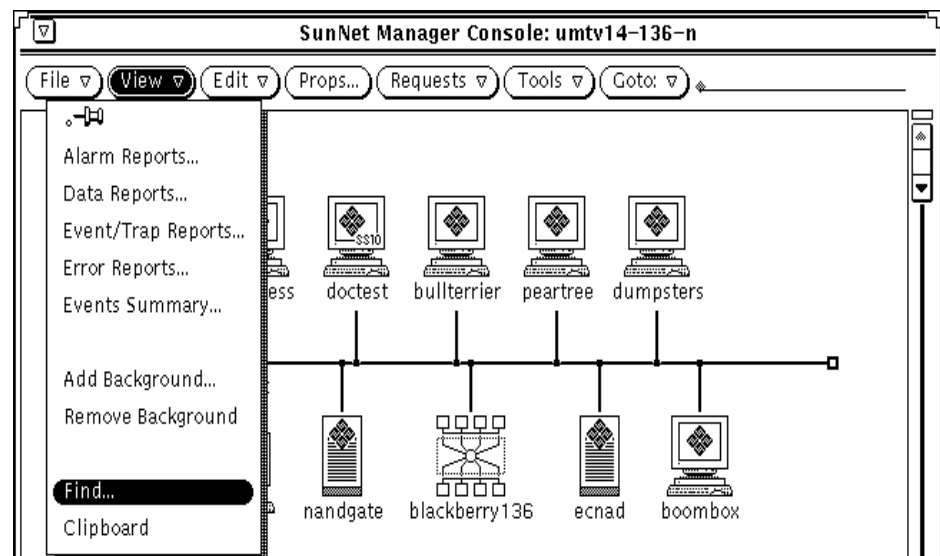


Figure 3-14 Selecting the View—Find Menu Item

2. In the Find window, enter the name of the element you want to find.
3. Click **SELECT** on the Find button in the window or press Return. The element is highlighted when found.
4. Click **SELECT** on the Next button to see the next view (if any) in which the element is located.

Use the Find function key on the left side of your keyboard to search on any of the names (aliases) associated with an element.

3.11 Modifying Element Properties

You can modify the following properties of an element instance in the element's Properties sheet:

- Agent schemas that apply to the element
- Element color
- Optional fields information

“Part 2: Reference” describes the fields in the Properties window. Note that you cannot modify the name of an element instance once it has been created.

To rename an element instance, you must delete and re-create it with the new name using the following steps:

1. Over the glyph that represents the element, press MENU. Or, click SELECT on the glyph and press the Props button on the left side of your keyboard.
2. Release MENU over Properties to open the Properties window for the element.

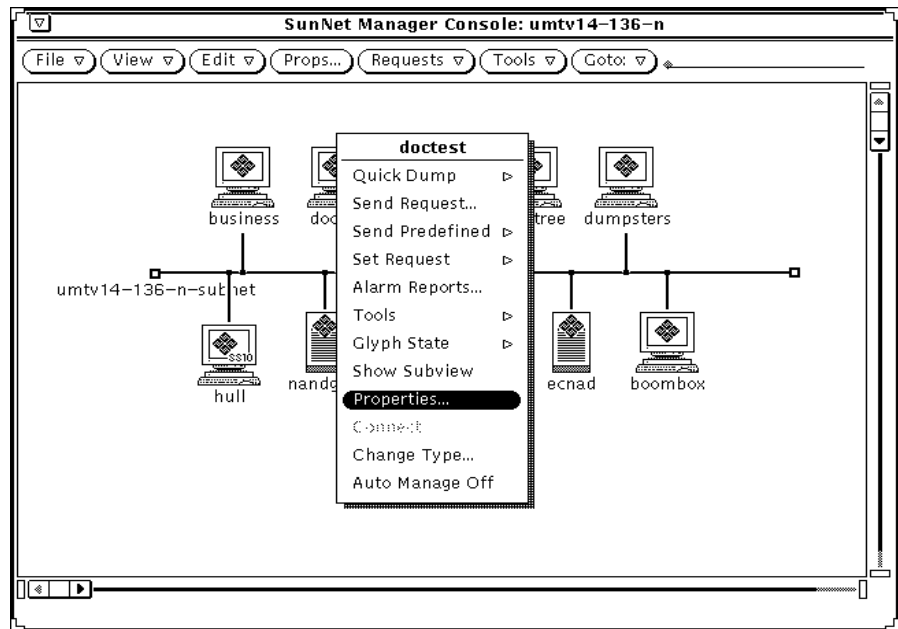


Figure 3-15 Selecting the Glyph—Properties Menu Item

3. Modify the fields within the Properties window for the element. Click SELECT on the Apply button to incorporate the changes to the element.

3.12 Changing Element Types

SunNet Manager has four categories of elements: component, view, bus, and connection, as shown in Figure 3-16.

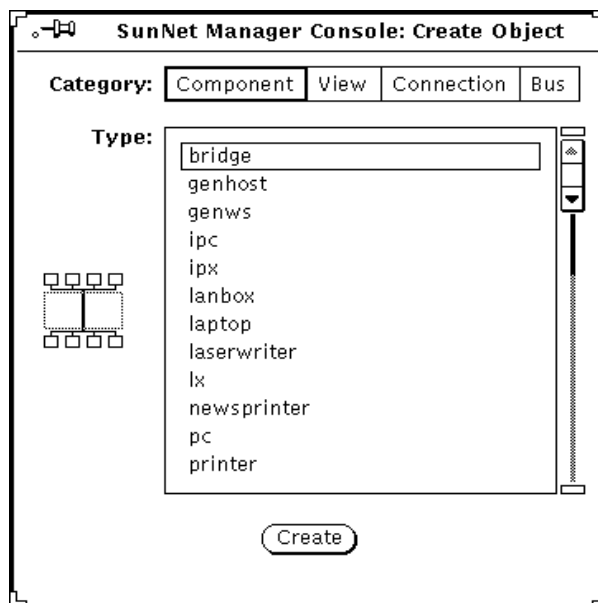


Figure 3-16 Element categories for Create Object

Of these categories, you can only change an element type in the component category. To change a component type, follow the steps below:

1. Over the glyph that represents the element that you want to modify. Press MENU.
2. Release MENU over Change Type. You receive the Change Type window shown in Figure 3-17.

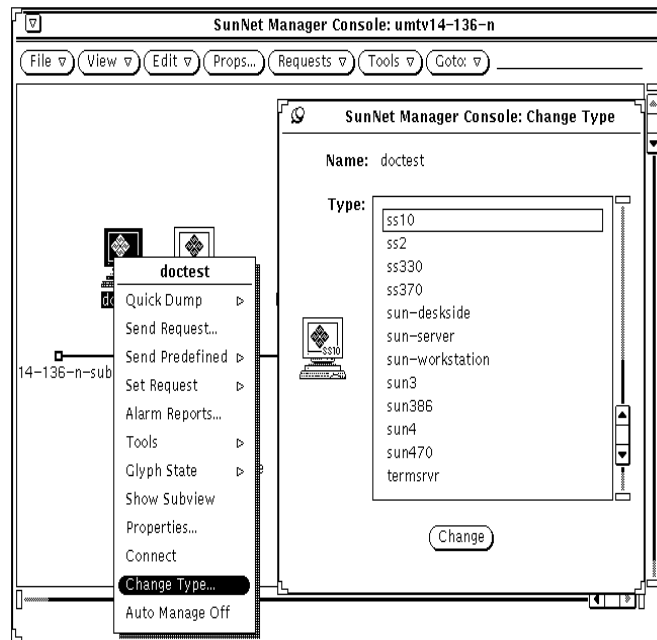


Figure 3-17 Change Type Window

3. Click **SELECT** on the new element type; click **SELECT** on **Change**.
The Change Type window disappears and the changed element appears in the Console window represented by a new glyph.

3.12.1 Restrictions on Changing Element Types

You can change element types for components, such as workstations, servers, printers. These types are displayed in the Type menu in the Change Type window, such as the one shown in Figure 3-17. You cannot change elements of category view, bus, or connection.

3.12.1.1 Elements on Different Machines

As the Discover tool finds network elements, it distinguishes among machines running SunNet Manager agents, SNMP devices, routers, and networks and subnetworks. (You can restrict the search to one or any combination of these types of objects.) The first three object types are in the component category of elements. The last, networks and subnetworks, are in the view category and cannot be changed.

3.13 Moving Elements within a View

You can move elements within a view as described in the following steps:

1. Over the glyph that represents the element to be moved, press **SELECT**. If you wish to move multiple elements at the same time, click **ADJUST** on subsequent glyphs.
2. Drag the mouse pointer to the desired new location and release. Figure 3-18 shows moving a glyph.

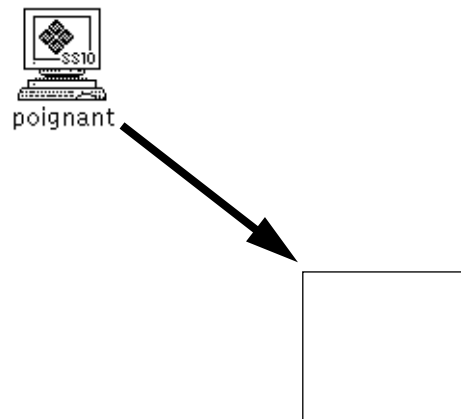


Figure 3-18 Moving a Glyph to a New Location

3.14 Moving Elements from One View to Another

You can move elements between views using the steps below:

1. **Over the glyph that represents the element to be moved, click SELECT.**
To move multiple elements at the same time, click ADJUST on subsequent glyphs.
2. **In the control area of the Console window, invoke Edit►Cut.**
Or, **press MENU over an empty space in the view to open the Edit menu as a floating popup menu.**
3. **Release MENU over Cut.**
4. **Use the Goto button in the Console's control area to switch to the view where the glyph(s) will be moved.**
See "Traversing the View Hierarchy" earlier in this Chapter for more information.
5. **In the Console's control area, invoke Edit►Paste.**

3.14.1 Useful Tips

1. You can use the Cut and Paste function keys on the left side of your keyboard instead of Edit menu functions.
2. When you cut or copy an element or set of elements, the element(s) is stored in the clipboard. To view the contents of the clipboard in the Console window, press MENU on View►Clipboard and release MENU.
3. You cannot cut an element with an associated request, nor delete an element that contains another element in its subview.

3.15 Connecting Elements

You can draw a connection from a selected (highlighted) element to the element glyph where the mouse pointer is located.

1. **Click SELECT over one of the elements (the element to connect from).**
2. **Press MENU over the other element (the element to connect to).**
3. **Release MENU over Connect.**

Figure 3-19 shows a connect example.

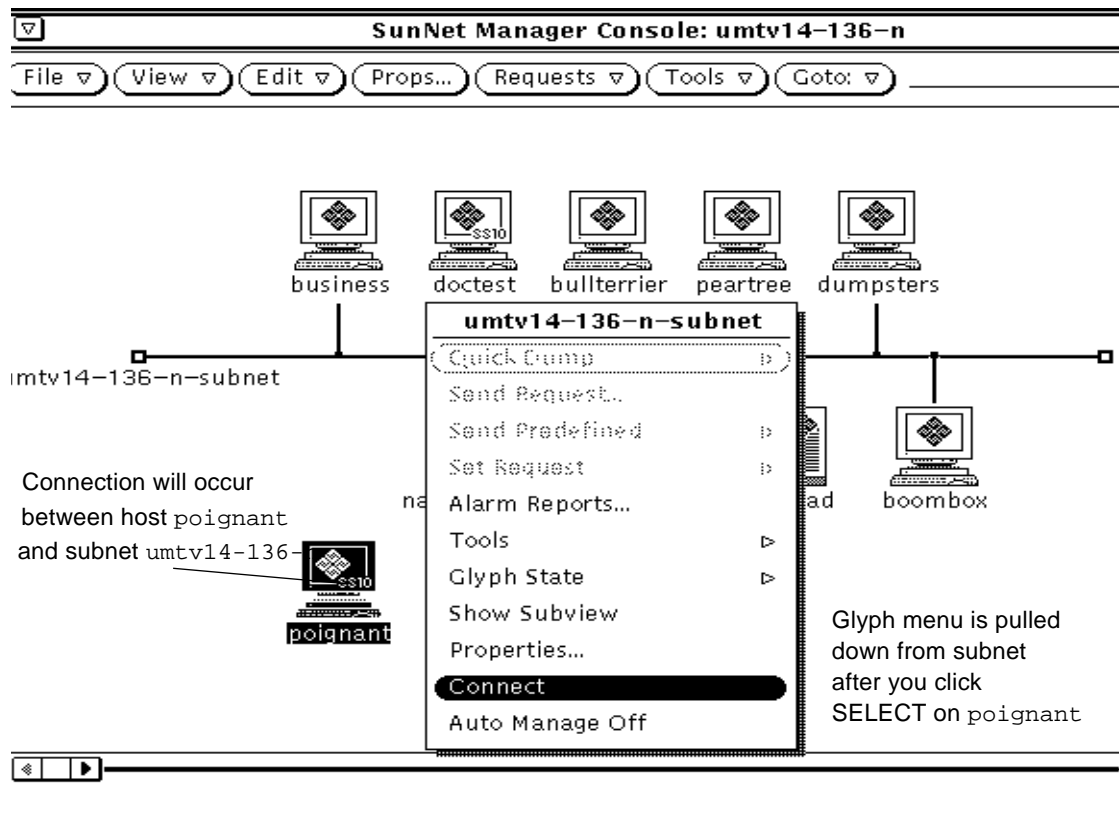


Figure 3-19 Glyph—Connect Example

3.15.1 Simple Connections

These connections should not be confused with the Connections element type category, which includes RS-232 connections and other links. The connections described in this task are also known as “simple” connections. They are *not* elements and cannot be managed. Simple connections are useful only for showing a graphic representation of the connectivity between elements.

When either of the connected elements is moved, the connection also moves.

The color of the connection is based on the color of the second, “connected to” element. Whenever you select (highlight) this element, the connection is also highlighted.

3.16 Copying Elements

After you have created an element, you can create many instances of that element in various views in your database. To do this, copy the element from one view to another as explained below.

- 1. Click SELECT on the glyph that represents the element to be copied to highlight it.**
If you wish to copy multiple elements in a view at the same time, click ADJUST on subsequent glyphs.
- 2. In the control area of the Console window, invoke Edit►Copy. Or, press MENU over an empty space in the view**

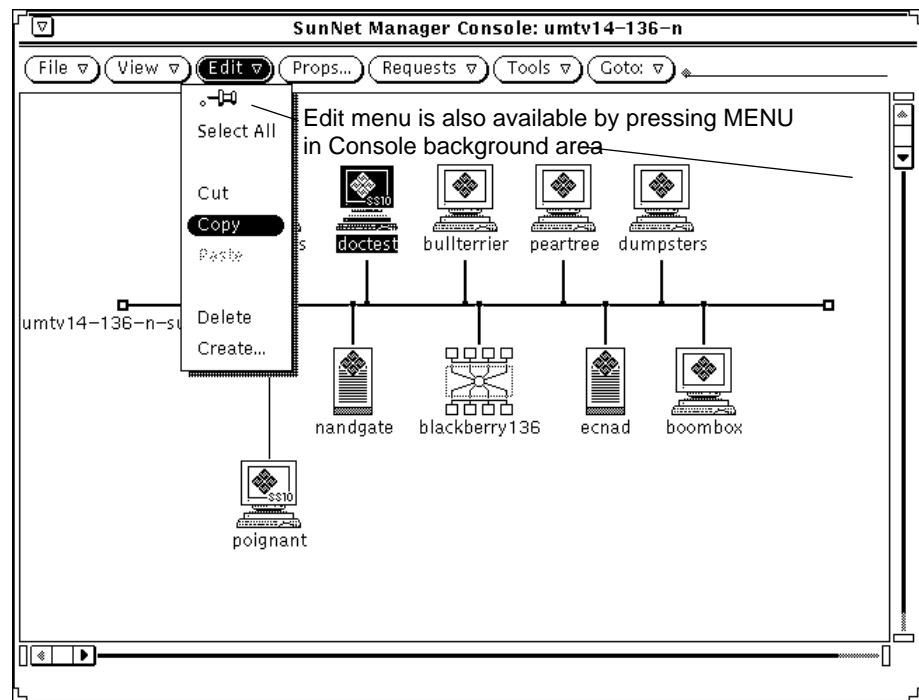


Figure 3-20 Selecting the Edit—Copy Menu Item

3. Use the Goto menu to move to the view to which the element will be copied.
4. Invoke Edit►Paste (from the Edit button or from a floating popup menu in an empty space in the new view).

When you cut or copy an element or set of elements, the element(s) is stored in the clipboard. To view the contents of the clipboard in the Console window, press MENU on View►Clipboard and release MENU.

3.16.0.1 Using Function Keys for Copy and Paste

You can use the Copy and Paste function keys on the left side of your keyboard instead of the Edit menu functions. The Copy function copies the element into the Console clipboard. After an element has been copied into the clipboard, it can be pasted into as many views as needed.

3.16.0.2 Drag and Drop

If an element and the view into which it will be copied are both displayed in the same view, you can use the Console's "drag and drop" function:

- 1. Press SELECT over the glyph that represents the element to be copied.**
- 2. While still pressing the SELECT mouse button, drag the mouse pointer to the glyph that represents the view to which the element will be copied. Release the mouse button.**

3.17 Deleting Elements

You can delete an element instance from a view using the steps below:

- 1. Click SELECT over the glyph to be deleted. If you wish to delete multiple elements at the same time, click ADJUST on subsequent glyphs.**
- 2. In the Console's control area, invoke Edit►Delete.Or, press MENU over an empty space in the view.**

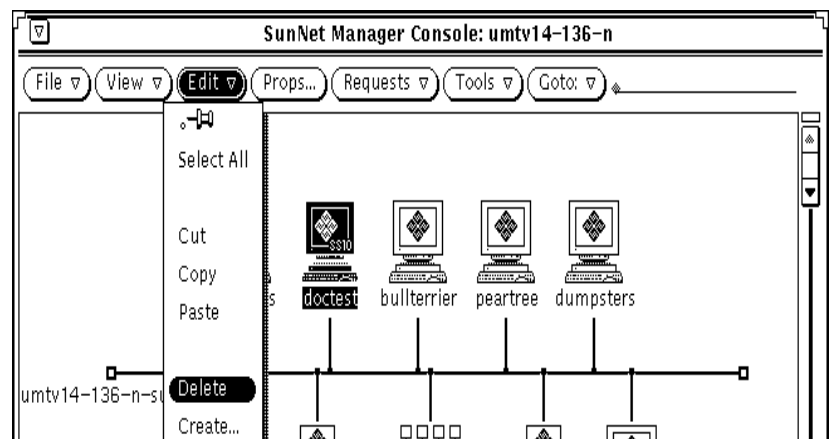


Figure 3-21 Selecting the Edit—Delete Menu Item

3.17.1 Useful Notes About Delete

You cannot delete an element that has a request associated with it. You also cannot delete an element that contains another element in its subview.

Delete removes the element from the view and from the database. The element is not placed in the clipboard, so you cannot paste a deleted element.

3.18 Saving the Management Database

The runtime database contains element and request instances that you have created. You can save the database to an ASCII file, allowing you to reinvoke the Console with the same database. Follow the steps below:

1. In the Console window, press **MENU** on **File>Save>Management Database** and release **MENU**.

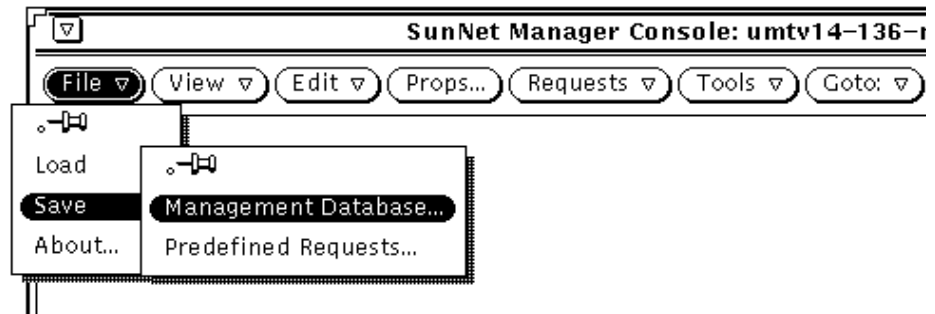


Figure 3-22 Selecting the File>Save>Management Database Menu Item

2. In the Save pop-up window, enter the directory and file name for the database.

3. Click SELECT on the Save button to store the database.

After you invoke File>Save>Management Database, SunNet Manager takes a snapshot of the current database and stores it in a single ASCII file. The structure of this stored database—a single file—is entirely different from the structure of the runtime database. The runtime database consists of a set of special database files under a directory name having the following form:

- /var/adm/snm (or \$SNM_HOME)/db.<login> (or db.<\$SNM_NAME>) for Solaris 1.1.1
- /var/opt/SUNWconn/snm (or \$SNM_HOME)/db.<login> (or db.<\$SNM_NAME>) for Solaris 2.x

3.18.0.1 Don't Store Under /Var

It is recommended that you not store the ASCII files under /var because /var is for files that grow and because a saved ASCII database is static and possibly large.

When restarting the Console, you can specify whether to continue to use the runtime database from the previous run of the Console, or to reinitialize the database from ASCII structure and instance files. See the section "Starting the Console" in this Chapter for instructions for reinitializing the database.

3.19 Quitting the Console

You quit a Console session by using the Quit option of the Console's window menu.

1. In the title bar of the Console window, press **MENU** and release **MENU** over **Quit**.
2. If there are any active requests, you see the pop-up verification window in **Figure 3-23**.



Figure 3-23 Console Quit Window

Click **SELECT** on one of the following:

- **Quit**, to have the Console kill all active requests before stopping.
- **View Requests**, to display the Request Viewer window.
- **Resume**, to continue running the Console.

When you quit a Console session, the verification window is displayed by default if there are any active requests. You can choose to have the verification window *always* displayed or *never* displayed—this is specified in the Console Properties window, under the Miscellaneous category. See the description of the Verify Quit setting in “Part 2: Reference.”

Requesting Data



This chapter discusses the following topics:

- Making a one-time request for data
- Requesting periodic data
- Copying requests
- Viewing incoming data
- Analyzing stored data
- Modifying the display of a graph
- Printing a graph

You can request that an agent return values for some or all attributes in a single agent group. Your request can either be for a single, quick snapshot of all the values in the agent group, or it can be a request for periodic reporting of one or more attribute values. For the latter type of requests, you can specify report characteristics such as:

- Number of reports from the agent
- Interval between reports
- Whether the reports should be stored in a file
- Whether the attribute values reported should be displayed graphically

This chapter describes how to request data reports from agents and how to view or analyze the returned data.

4.1 Making a One-Time Request for Data

A Quick Dump retrieves all the values in an attribute group for a target system. You can initiate a Quick Dump request in one of two ways.

4.1.1 Quick Dump using the Console Requests Menu

1. Click **SELECT** over the glyph that represents the target element.
2. Press **MENU** on the Requests button in the SNM Console.
3. Release **MENU** over **Quick Dump** to receive a Request Builder window such as the one shown in Figure 4-1.

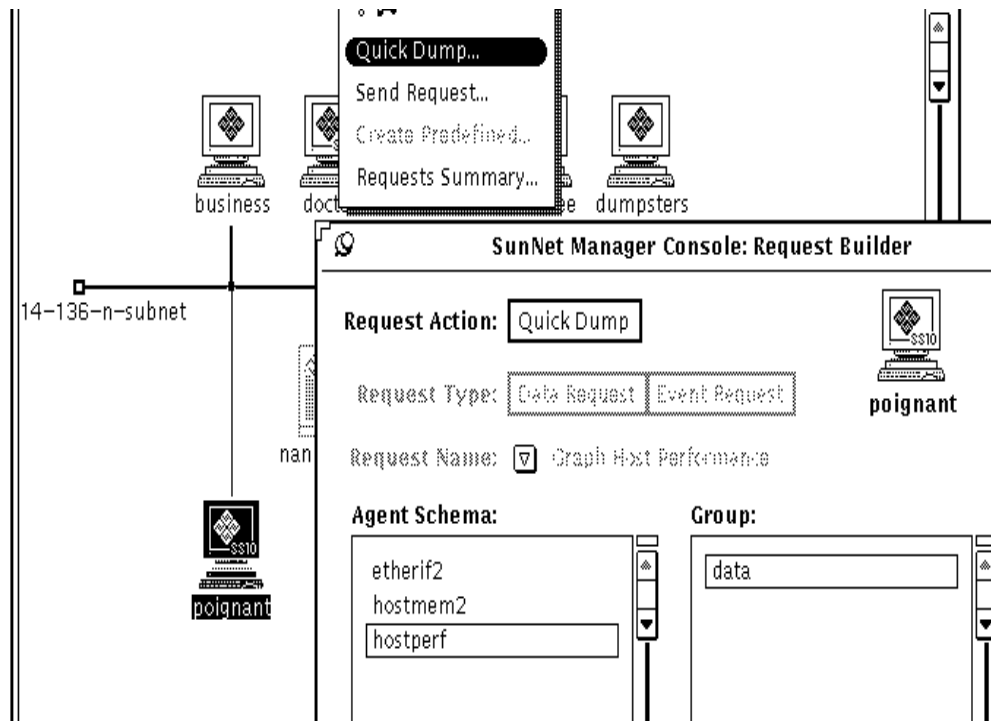


Figure 4-1 Requests Menu—Quick Dump Request

4. In the Request Builder window, click **SELECT** on the Agent Schema, then on the Group you want for that schema. Click **SELECT** on Apply to send the request. You receive a Quick Dump Report window, as shown in Figure 4-3.

4.1.2 Quick Dump Through the Glyph Menu

1. Press **MENU** over the glyph that represents the target element.
2. Press **MENU** over Quick Dump and drag to the right over the desired agent name and attribute group name as shown in Figure 4-2.

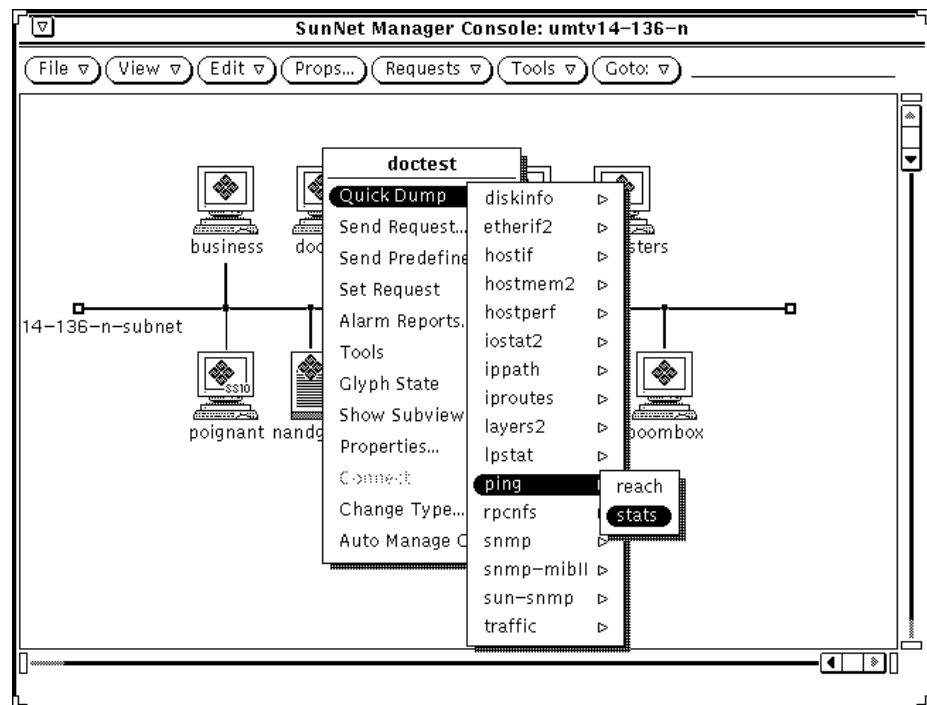


Figure 4-2 Glyph Menu—Quick Dump Request

Releasing **MENU** on the attribute group name starts the Quick Dump request. You receive a Quick Dump Report window, as shown in Figure 4-3.

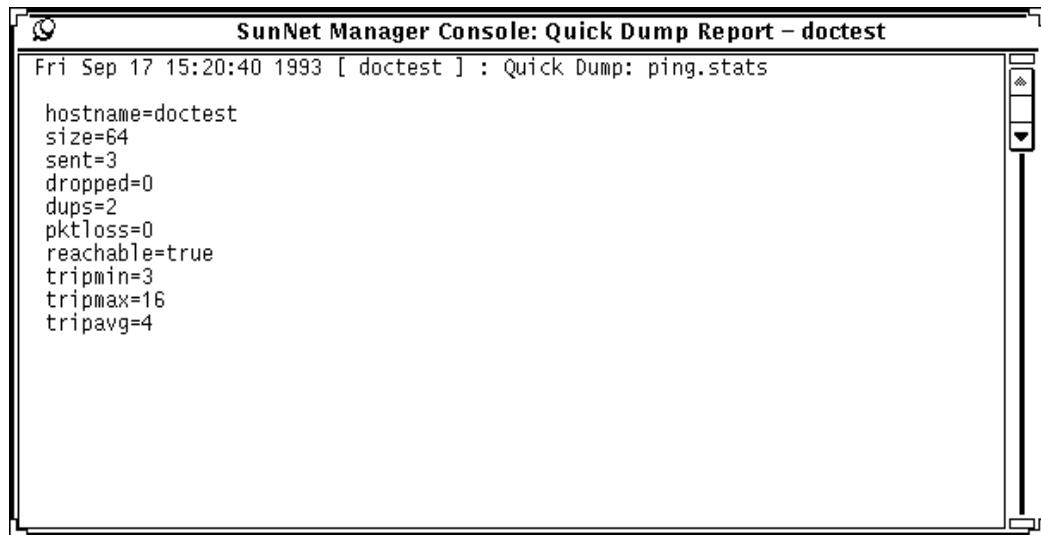


Figure 4-3 Quick Dump Report Window

4.1.3 Useful Notes About the Quick Dump Window

Starting with version 2.3, a popup window is displayed and filled with information as it is gathered from the appropriate managed devices.

Table attributes are displayed in tabular form when possible and the Quick Dump window is automatically stretched if necessary.

Data returned in the Quick Dump window is not stored and is not displayed in the Data Reports window.

Starting with the current product, the pop-up window appears as soon as the quick dump request is submitted. Results of the request display when they are returned from the agent. If an error occurs, the message “cannot start request *<request name>*” displays in the pop-up window. The text of the error can be displayed through the View►Error Reports window.

4.1.3.1 *Quick Dump Information from Set Tool*

The Get operation of the Set Tool returns the same information for SNMP devices as the Quick Dump. The Set Tool is described in “Part 2: Reference.”

4.2 *Making a Request for Periodic Data*

A data request causes an agent to automatically report the values of specified attributes for a target system at specified intervals. You can compose your own request, targeting a specific machine for a specific set of information. More conveniently, you can use one of the predefined requests. These requests save you the trouble of composing requests for individual machines.

4.3 *Prioritizing Requests*

Starting with the current version of SunNet Manager, you can prioritize data requests by specifying:

- Start time
- Start date
- Stop time
- Stop date

This is useful, for example, if you want a request to begin when no one will be available to initiate it, or when network traffic is at a minimum. The default is to start immediately and run indefinitely. If you enter a Stop time, it takes precedence over any count value that has been specified. A count value of 0 means “run indefinitely.” If a count value is reached before Stop time, the request would stop. To initiate the prioritizing process:

1. Define your request in the Request Builder window, then click SELECT on Apply. You receive the screen shown in Figure 4-4.

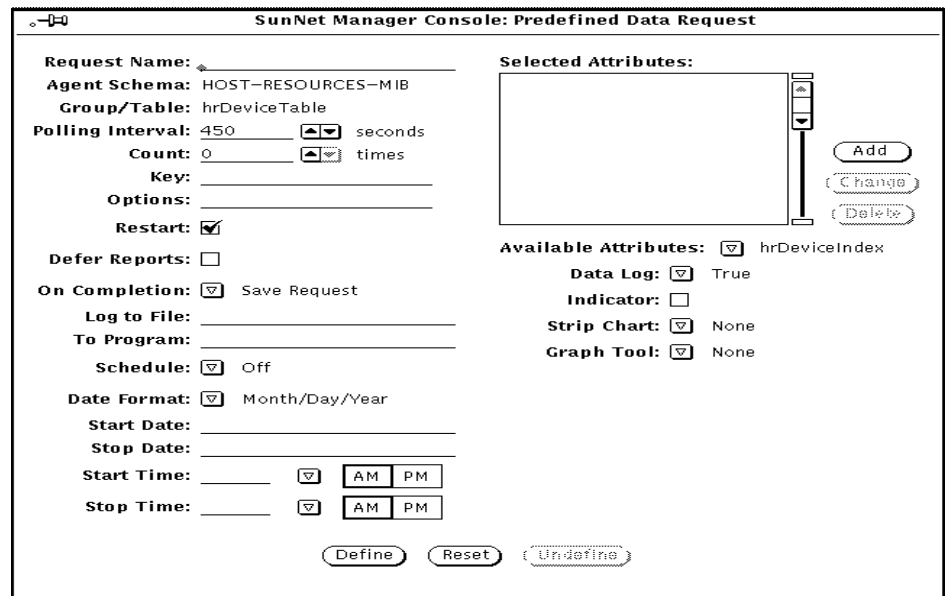


Figure 4-4 Request Schedule Menu

2. Follow the prompts on the Request Builder Menu to specify start and stop for date and time.

4.3.1 Sending a Data Request Through the Console Requests Menu

1. Move the mouse pointer over the glyph that represents the target element, and click SELECT.
2. Press MENU on the Requests button, and release MENU over Send Request.
You will receive the window in Figure 4-5. The Request Action field is automatically set to Send Request.

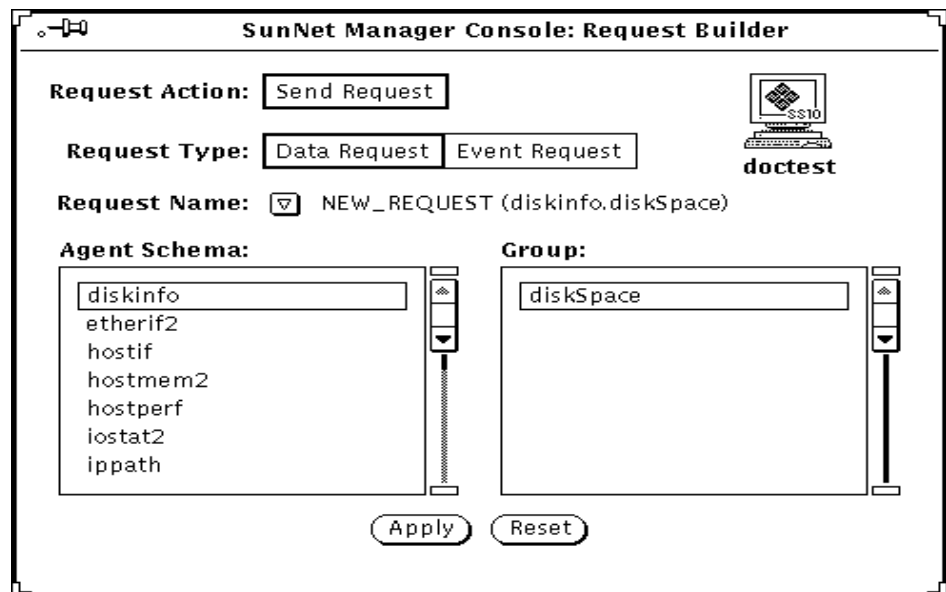


Figure 4-5 Sample Request Builder Window

3. Click **SELECT** on the **Request Type** you want to send (**Data**, in this example).
4. Press **MENU** on the **Request Name** abbreviated menu button. You receive the menu shown in Figure 4-6.

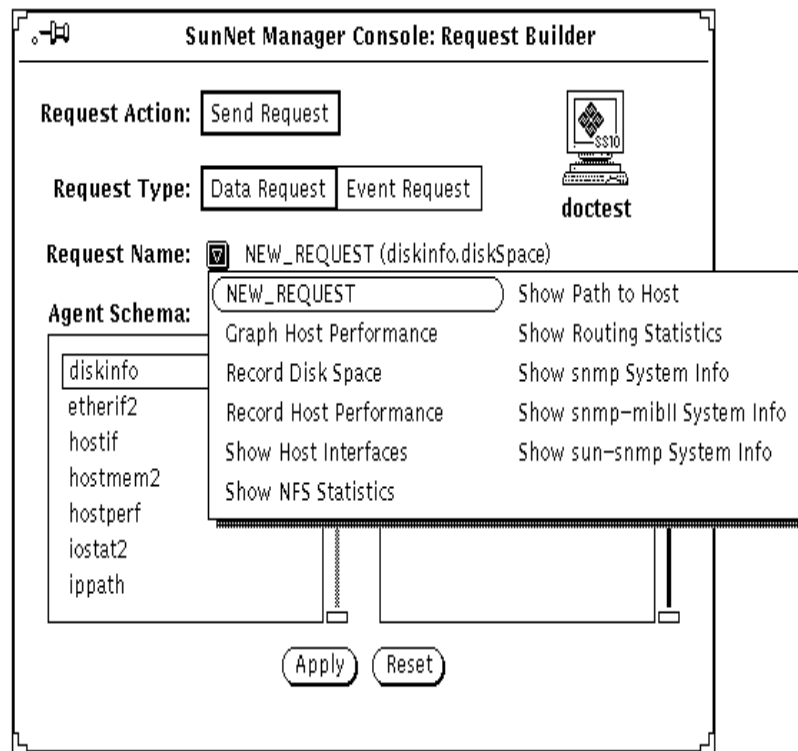


Figure 4-6 Request Name Menu

5. Choose a request from the Request Name menu.
6. After choosing either NEW_REQUEST or a predefined request name from the Request Name menu, click SELECT on the Apply button. You receive a Data Request Properties window as shown in Figure 4-7.

4.3.2 *Creating Your Own Request*

There can be times when you want to create your own request. You can accomplish this using the following steps:

1. **Click SELECT on an agent in the Agent Schema scrolling list.**
2. **Click SELECT on a group name in the Group scrolling list.**
 - b. **Choose NEW_REQUEST from the Request Name pull-down menu.**
3. **After choosing either NEW_REQUEST or a predefined request name from the Request Name menu, click SELECT on the Apply button. You receive a Data Request Properties window as shown in Figure 4-7.**
4. **Examine the Request Properties window and make any changes you want.** Many predefined requests will require little or no modification.

SunNet Manager Console: Predefined Data Request

Request Name: Graph Host Performance
 Agent Schema: hostperf
 Group/Table: data
 Polling Interval: 30 seconds
 Count: 0 times
 Key:
 Options:
 Restart:
 Defer Reports:
 On Completion: Save Request
 Log to File:
 To Program:
 Schedule: Off
 Date Format: Month/Day/Year
 Start Date:
 Stop Date:
 Start Time: AM PM
 Stop Time: AM PM

Selected Attributes:
 cpu%
 intr
 disk
 ipkts
 opkts
 Add
 Change
 Delete

Available Attributes: cpu%
 Data Log: True
 Indicator:
 Strip Chart: None
 Graph Tool: Absolute Values

Define Reset Undefine

Request Name: Graph Host Performance

Figure 4-7 Sample Data Request Properties Sheet

The example above shows the predefined data request “Graph Host Performance.”

5. On the left side of the request window, modify or fill in the report characteristics fields.

See the Section below on “Scheduling Requests” for more information.

6. On the right side of the request window, select (or accept) the attributes on which you want the agent to send data, and how you want to view the data.

7. Press MENU on the Attribute menu button and release MENU over the attribute you want.

8. Click **SELECT** on **Apply** to add the attribute to your request. For a description of the fields in the Data Request template, refer to “Part 2: Reference.”
9. Click **SELECT** on the **Start** button to send the data request to the agent.

4.3.3 Sending a Data Request Using the Glyph Menu

1. Move the mouse cursor over the glyph and pull down the Glyph menu.
2. In the Glyph menu, release **MENU** over **Send Request**. You then receive a **Request Builder** window, from which you can use a predefined request or create your own request.

4.3.4 Sending a Predefined Data Request Through the Glyph Menu

SunNet Manager is shipped with a number of predefined data requests. Use these as a convenient way to request routine data reports. To send such a request using the Glyph menu, follow the steps below:

1. Move the mouse pointer over the glyph for the element to which you want to send the request.
2. Press **MENU** at **Send Predefined**.
3. Move the mouse pointer to the right over **Data Request** and continue to pull right to obtain the menu of predefined data requests shown in **Figure 4-8**.

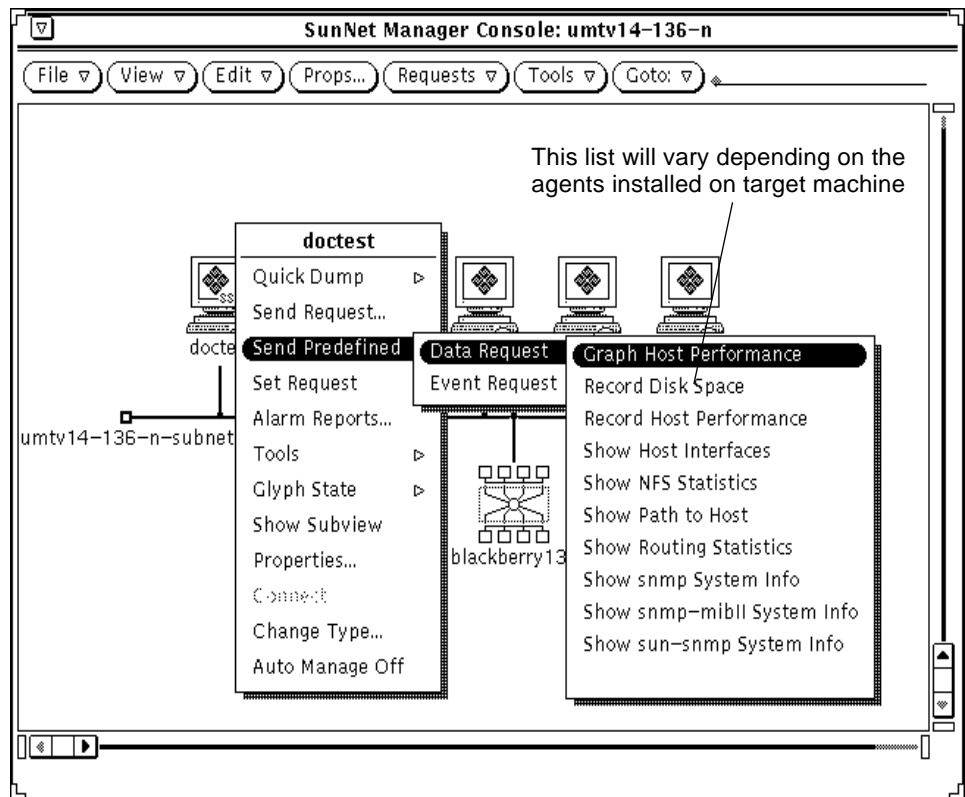


Figure 4-8 Predefined Data Request

4. Release MENU over the predefined request you want.

You receive no Properties window for the request; the request is sent immediately following your menu selection.

To view the properties associated with a predefined request:

1. Click **SELECT** on the glyph to which you are sending a request.
2. Invoke **Requests**►**Send Request**.
3. In the **Request Builder** window, choose the name of the predefined request you want from the **Request Name** menu.

4. Click SELECT on Apply. You receive the Properties window for that predefined request.

Alternative ways to view the properties of a request:

- Double-click on the glyph for the target of your request.
- Click SELECT on the glyph for the request.
- Click SELECT on the Props button in the Console's control area.
- Invoke Requests ► Requests Summary.
- Find and click SELECT on your request.
- Click SELECT on the Props button in the Requests Summary window.
- Use the Select and Sort menus to find your request.

See “Viewing and Modifying Properties of a Request” on page 4-35 for more information.

For further background on predefined requests, see “Part 2: Reference.”

4.3.5 Predefined Requests Depend on Agents Present

Predefined requests (and the agents listed in the Agent Schema list in a Request Builder window) for a given machine depend on the agents that have been checked off as present in the properties window for that machine.

A machine the Discover tool adds to your database which does not have SunNet Manager agents on it, has the `hostperf` and `ping` agents checked off. Information is available from a machine without agents through the proxy feature, just as if the `hostperf` and `ping` agents *were* on that machine. A machine with agents installed that the Discover tool adds to your database, has the agents checked off that are actually present.

4.3.6 Notes About Console Window Features

Messages indicating that the request has been started are displayed in the lower left footer of the Console window.

The default value of the Restart field in the Request Properties window is specified in the Requests category of the Console's Properties window. You access the Console's Properties window by clicking SELECT in the Props button in the Console's control area. For more information, see "Part 2: Reference."

4.4 Copying Requests

Copying requests allows you to quickly start the same Data Request for multiple elements. Each request is represented by a glyph that appears in the subview of an element. Because requests are represented by glyphs, copying requests is similar to copying elements.

1. **Move the mouse pointer over the source element. Double-click SELECT to display the subview of the element.**
2. **Move the mouse pointer over the request glyph you want to copy, and click SELECT.**
3. **Press MENU over the Edit button to open the Edit menu.**
4. **Drag the mouse pointer down to the Copy option and release.**
5. **Move the mouse pointer over the target element (in either the same view or a different view). Double-click SELECT to display the subview of the element.**
6. **In the element's subview, press MENU to open the Edit menu. Drag the mouse pointer down to the Paste option and release.**

4.4.1 Using Function Keys for Copy and Paste

You can use the keyboard function keys that correspond to Copy and Paste instead of the Console Edit menu.

The Copy function copies the request into the clipboard. Once a request has been copied into the clipboard, it can be pasted into as many views as needed. In the Console window, invoke View►Clipboard to view the contents of the clipboard.

Pasting a request into the view of an element launches the request for that element. If you specified a name for the original request, the same name is used for the copied request. If you did not specify a name for the original request, the request name assigned to the copied request is incremented by 1. For example, if the original request uses the name `hostperf.data.0`, then a copied request becomes `hostperf.data.1`.

As an alternative to copying a request, you can use the predefined request feature, described in this chapter under “Sending a Predefined Data Request through the Glyph Menu.”

4.5 Viewing Incoming Data

You can view Data Request results as they are received by the Console in one of the following ways:

Data Reports

The Data Reports option of the View menu displays data reports in a simple text format. The values of attributes specified in a single Data Request are displayed.

Results Grapher

The Results Grapher displays the values of a specified attribute on a graph. The Grapher is intended to aid in the visual analysis of collected attribute information. You can modify the display of graphs, including specifying graph colors and two-dimensional or three-dimensional displays. Graphs can be merged for comparison of data.

Strip Chart

A Strip Chart can also display the values of a specified attribute in a simple graph. A Strip Chart is a simple graph of a single attribute for a single system. Although Strip Charts can be copied and pasted into different views, you cannot merge multiple Strip Charts. Because each Strip Chart is automatically scaled, it is not very practical to try to compare data among multiple Strip Charts.

An Indicator displays the last reported value of an attribute for a particular request. If many different Data Requests are active, an Indicator can be a convenient way of seeing the latest reported value of an attribute.

4.5.1 Viewing Incoming Data for All Systems: Data Reports Window

To see attribute values sent by the agent in response to a Data Request, follow the steps below:

1. In the Console window, press **MENU** in the View menu and release **MENU** over Data Reports as shown in Figure 4-9. (Or, click **SELECT** over the target glyph, and invoke **View** and **Data Reports**.)

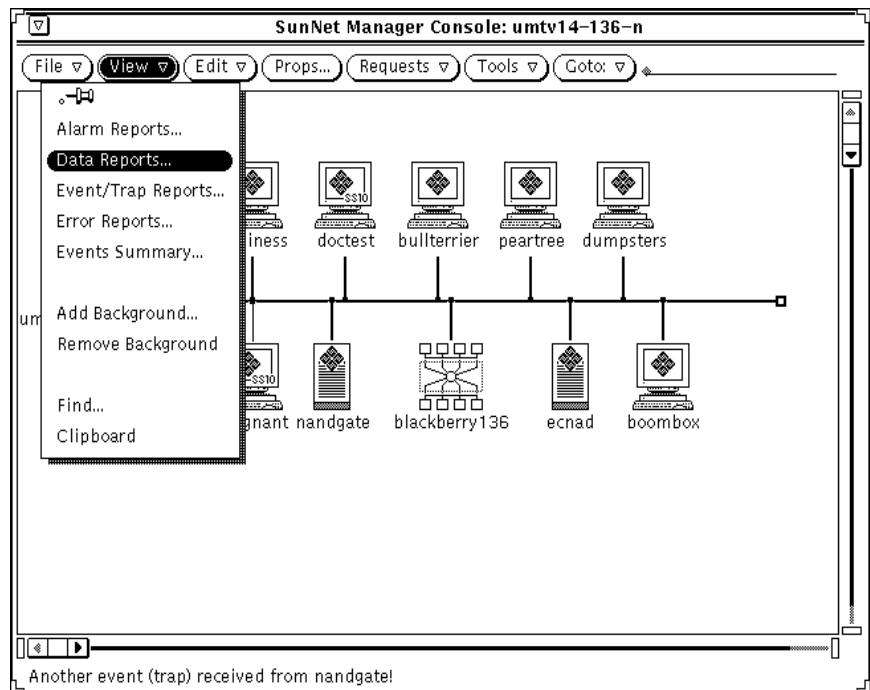


Figure 4-9 View—Data Reports Option

You receive the window shown in Figure 4-10.

2. To examine data reports for a particular system, type in the device name after the **Device** prompt and press **Return**.
3. To see entries for all elements again, press **Ctrl-U** to clear the **Device** line and press **Return**.

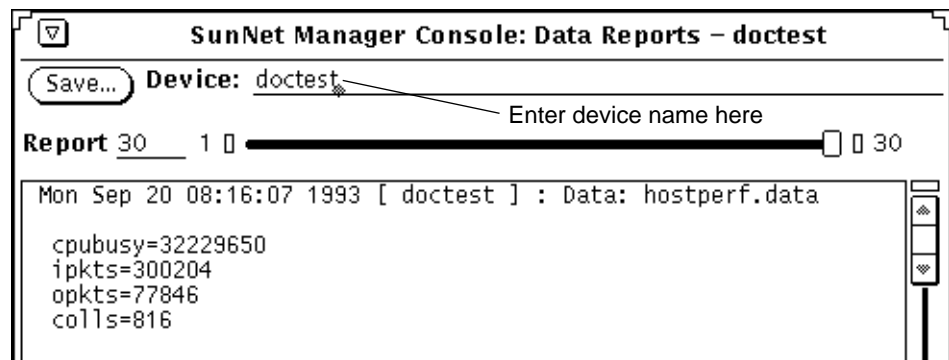


Figure 4-10 Sample Data Reports Window

4. To save the data reports to a file, click SELECT on the Save button.

A pop-up Save window appears, where you can enter the path and file name where the data reports are to be stored.

5. To browse through entries in the window:

- Press SELECT on the slider and drag the slider to the left or right.
- Click SELECT to the left or right of the slider.
- Click SELECT on either end of the slider bar.
- Enter the number of a report on the Report line and press Return.

The Data Reports window contains the most recently received data reports for all systems. The currently displayed entry and the total number of entries are identified in the slider bar. When Device is set for a particular element, the slider bar reflects the current and total number of entries for the element.

By default, the maximum number of data reports displayed in the Data Reports window is 1000. When the maximum limit is exceeded, the oldest data report is deleted.

4.5.2 Viewing Incoming Data: Grapher

The Graph tool (also known as the Grapher) displays reported attribute values in a graph that you can modify.

1. Open the Data Request Properties window for the target element by either defining a new Data Request or modifying an existing request. The Data Request Properties window as shown in Figure 4-7 on page 4-10 will be displayed.
2. Move the mouse pointer over the Graph Tool abbreviated menu button and press MENU. Drag and release on either Absolute Values or Delta Values.

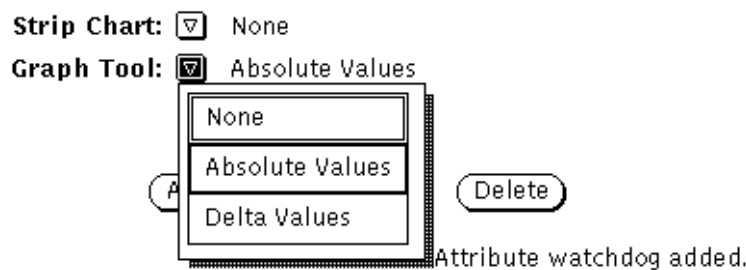


Figure 4-11 Graph Tool Menu

Choose Absolute Values to graph the values as they are reported. In graphs that use absolute values, the graphs use actual numbers, such as packet counts or collisions.

Choose Delta Values to graph the delta (or difference) between a previously reported value and the current reported value. In graphs that use delta values, the graph starts at zero or an arbitrary number and increments by however much it needs to accommodate changes.

Note – The predefined requests shipped with SunNet Manager have Absolute Values selected in the Graph Tool field.

4.5.3 Viewing Incoming Data: Strip Charts and Indicators

You can choose the Strip Chart or Indicator options in the Data Request Properties window. These options allow you to monitor the value of a single attribute from the Console view. An Indicator shows the last reported value of an attribute. A Strip Chart is an auto-scaled chart of reported values for a single attribute. See Figure 4-12.

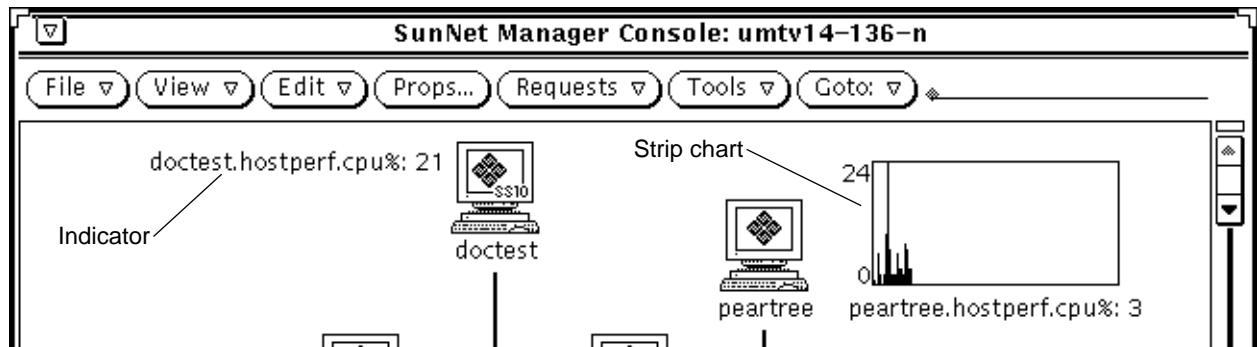


Figure 4-12 Strip Chart and Indicator Samples

1. **Open the Data Request Properties window for the target element by either defining a new Data Request or modifying an existing request. See Figure 4-13.**

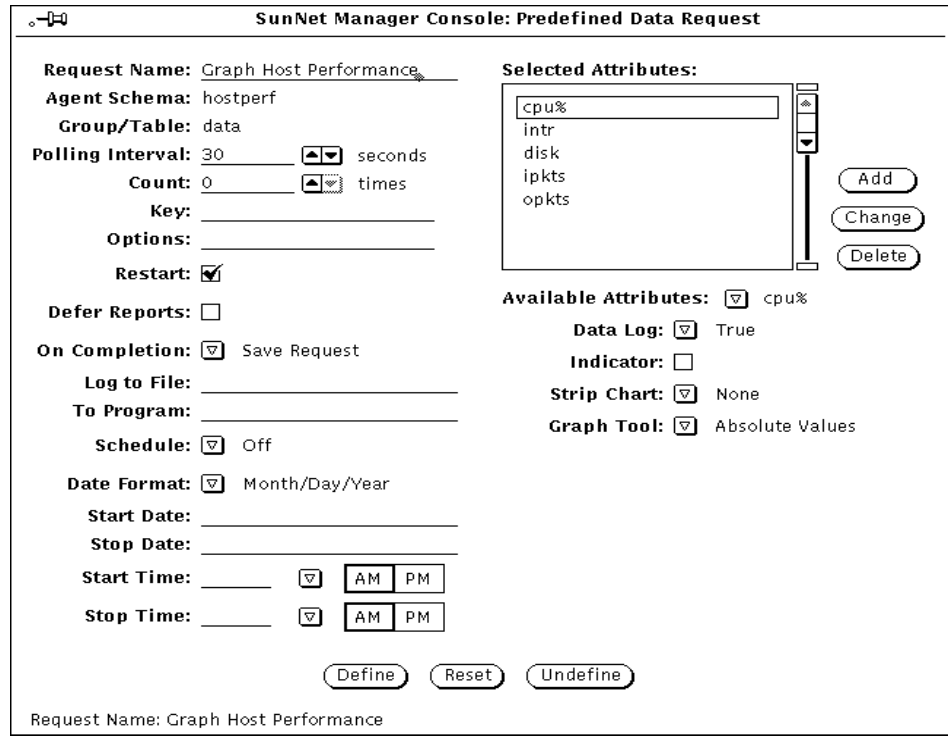


Figure 4-13 Data Request Properties Template

2. To specify that the attribute data be displayed in an Indicator, move the mouse pointer over the box next to the Indicator field and click SELECT. A check mark appears in the box.
3. To specify that the attribute data be displayed in a Strip Chart, move the mouse pointer down over the Strip Chart abbreviated menu button and press MENU. Drag the mouse pointer down to either Absolute Values (chart the received values) or Delta Values (chart the differences between received values).

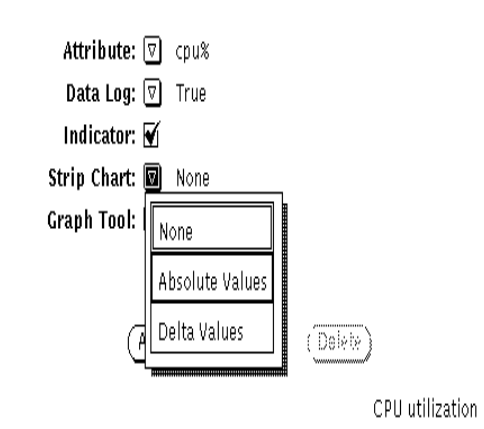


Figure 4-14 Strip Chart Menu

4.5.4 Useful Notes About Indicators and Strip Charts

Indicators and Strip Charts are automatically positioned in the view from which the request is launched. You can move Indicators or Strip Charts within a view, or cut, paste, or copy them into other views.

A maximum of 64 Indicators or 128 Strip Charts can be displayed in the view where the associated request is started.

You can modify certain properties of Strip Charts with their Properties window. For example, you can specify the maximum height and width of a Strip Chart. To modify the properties of a Strip Chart:

1. **Move the mouse pointer over the Strip Chart you want to modify and press MENU to open its Glyph menu.**
2. **Drag the mouse pointer down to Properties and release to open the Properties window.**
3. **Modify the Properties window, as necessary.**
Note that the Properties window for Strip Charts is similar to an Element Properties window. The middle portion of the window, for selecting agent schemas, and the bottom portion, for color selection, do not apply to Strip Charts.
4. **Click SELECT on the Apply button to modify the Indicator or Strip Chart.**

4.6 Analyzing Stored Data

You can analyze data that has been stored into a disk file by using the following tools:

- Results Browser
- Results Grapher

The following types of data files can be analyzed:

- Event/Trap Reports file. Unless a different log directory was specified during installation, this file is located at:
 - `/var/adm/snm/event.log` for Solaris 1.x installations.
 - `/var/opt/SUNWconn/snm/event.log` for Solaris 2.x installations.The log file is defined by the `event-log` keyword in the `/etc/snm.conf` file for Solaris 1.x installations and in the `/etc/opt/SUNWconn/snm.conf` file for Solaris 2.x installations.
- Data report entries that have been saved to a disk file (see the section “Viewing Incoming Data for All Systems: Data Reports Window,” in this chapter).
- Data report files specified with the Log to File field in the Data Request window.
- Any other files that use the data format specified by `snm.logfile(5)`.

4.6.1 Analyzing Stored Data: Results Browser

Use the Results Browser to retrieve and organize stored data.

1. **Move the mouse pointer over the Tools button on the Console Menu and press MENU to open the Tools menu. Or, invoke the Browser from the command line:**

```
host% snm_br.
```

2. **Drag the mouse pointer down to Browser and release to open the Results Browser. You receive the window in Figure 4-15**

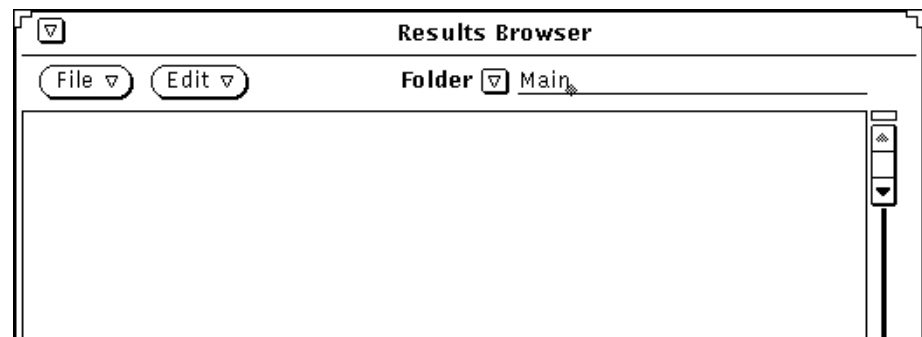


Figure 4-15 Results Browser Window

3. In the Results Browser file menu, move the mouse pointer over the File button and press MENU to open the File menu. Drag the mouse pointer down to Load. You receive the window in Figure 4-16.

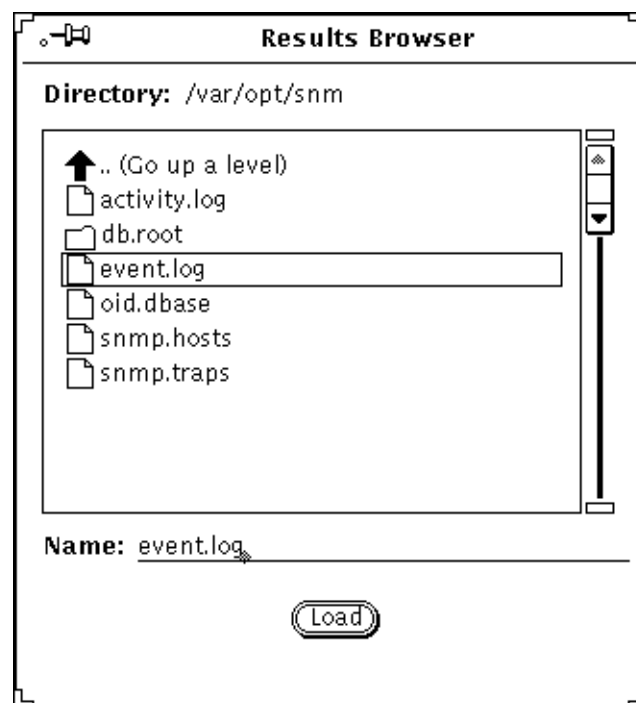


Figure 4-16 Results Browser File Menu

- 4. Enter the directory and file name of the file to be loaded and click SELECT on the Load button. You receive the window shown in Figure 4-17.**

Data loaded into the Browser is automatically organized into report streams and grouped by request and type of data. A *stream* is the set of requests associated with a specific pairing of hostname or IP address and agent name. For example, all of the data reports from the `hostmem2` agent for `poignant` would make up a single stream. The reports from the `layers2` agent for the same machine make up a different stream, as do the `layers2` reports for a different machine.

Name	Type	Protocol	Data Type	Time	Size
boombox	hostperf	data	Data	Sep 24 08:06:33	416
business	hostperf	data	Data	Sep 22 10:02:33	264
doctest	layers2	icmp	Data	Sep 22 09:47:39	40
doctest	rpcnfs	client	Data	Sep 24 08:15:12	20
eci_129.144.31.68	hostperf	data	Data	Sep 24 07:34:02	496
peartree	hostperf	data	Data	Sep 22 09:51:50	26
po_129.144.31.77	hostmem2	streams	Data	Sep 24 08:00:04	56

File performance.txt loaded, 1318 reports read.

Figure 4-17 Results Browser with Reports Loaded

5. To view individual reports in a report stream, double-click **SELECT** on the line that represents the stream.

The Browser runs as a separate process from the Console and remains running when you quit or close the Console.

The first column in the Browser scrolling list of streams can contain a hostname or an IP address.

4.6.1.1 *Printing Browser Reports*

You can invoke the Browser from a command line by entering `snm_br`. When you invoke the Browser from the command line, you can also specify the path and name of one or more data files to be loaded automatically when the Browser's main window is displayed. You can print Browser report streams to a specified printer. To do so:

- 1. Select the report streams you wish to print.**
 - a. Move the mouse pointer into the top portion of the Browser window where the names of the report streams are displayed.**
 - b. Press MENU to open the Streams menu.**
 - c. Drag the mouse pointer down to Select, then continue to drag right to By System, By Agent and Group, or By Report Type.**
 - d. Continue to drag right on the appropriate system name, agent group name, or report type.**
 - e. Repeat steps a through d until all desired streams are selected.**
- 2. Move the mouse pointer over the report streams and press MENU to open the Streams menu.**
- 3. Drag the mouse pointer down to Print. (See Figure 4-18.)**

The default printer is `lp`. To change the printer, modify the Browser Properties window as explained in the following summary.

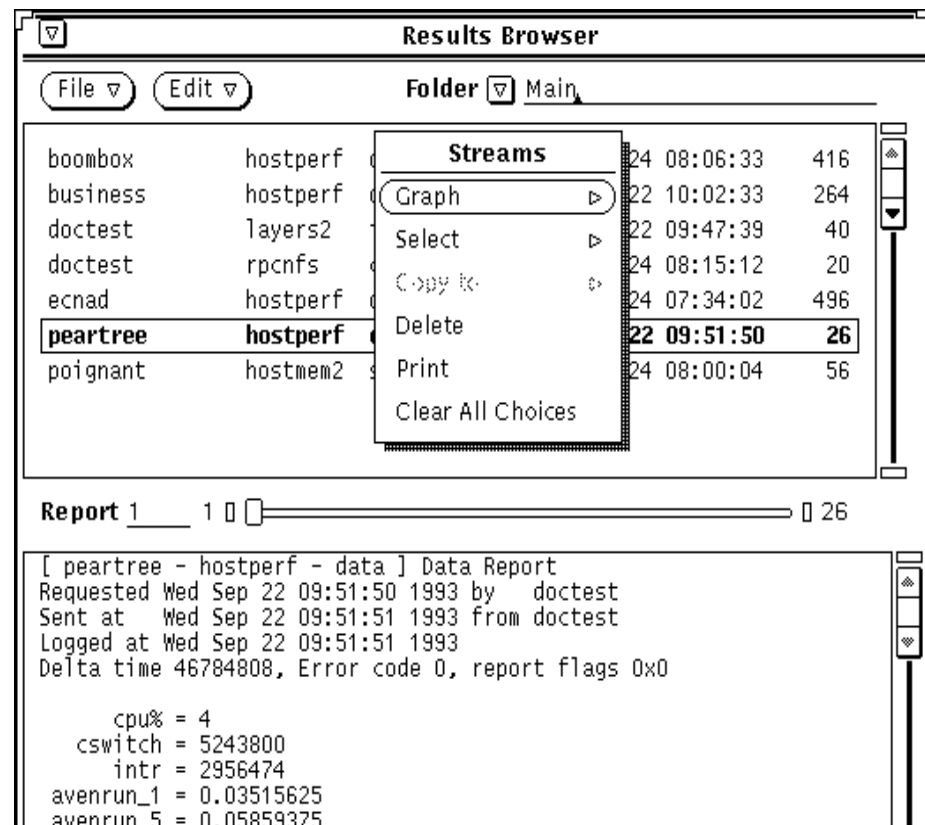


Figure 4-18 Browser Edit Menu

You can modify certain characteristics of the Browser display and the default printer. To do so:

1. Move the mouse pointer to the Edit button of the Browser window and press MENU to open the Edit menu.
2. Drag the mouse pointer down to the Tool Properties option and release. (See Figure 4-19.)

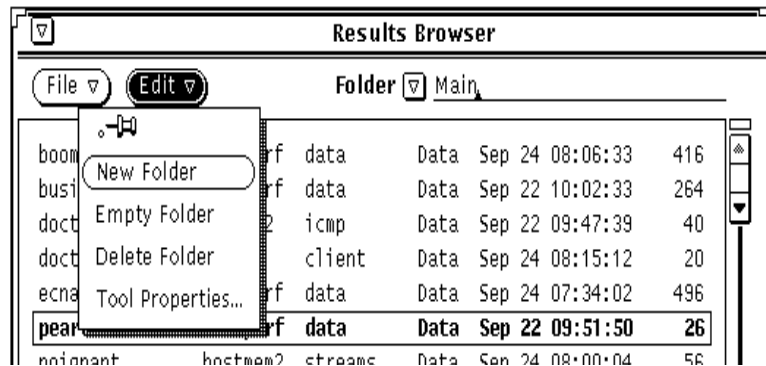


Figure 4-19 Browser Tool Edit Menu

3. Modify Browser display or print options. (See Figure 4-20.)

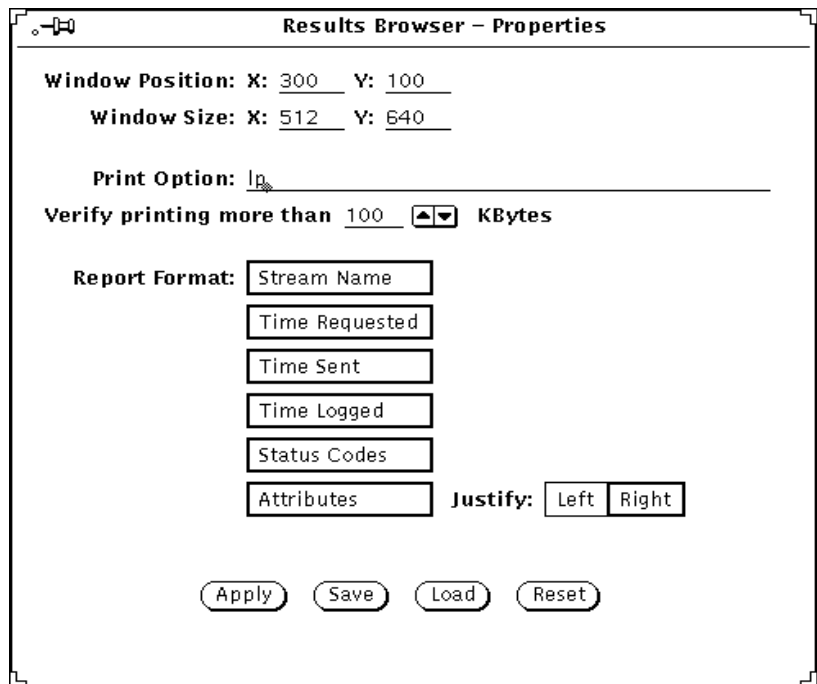


Figure 4-20 Browser Properties Window

For a complete description of Results Browser features, see “Part 2: Reference.”

4.6.2 Analyzing Stored Data: Results Grapher

The Results Grapher displays reported attribute values in a graph that can be manipulated.

You can invoke the Grapher directly, by selecting Tools►Grapher in the Console window or by invoking `snm_gr` on a command line.

However, it is often more useful to invoke the Grapher from the Browser, then, enter the attribute name you want. The procedure for this is specified below.

1. **Move the mouse pointer over the Tools button and press MENU to open the Tools menu. Drag the mouse pointer down to Browser and release. Or, invoke the Browser from the command line:**

```
host% snm_br
```

2. **Load the file into the Results Browser.**
 - a. **Move the mouse pointer over the File button and press MENU to open the File menu. Drag the mouse pointer down to Load.**
 - b. **Enter the directory and path name of the file to be loaded and click SELECT on the load button.**
3. **In the Browser window, send the stream(s) that contain the data to be graphed to the Grapher.**
 - a. **Select the stream that contains the data to be graphed.**
 - b. **Move the mouse pointer into the scrolling list in the upper portion of the Browser window and press MENU to open the Streams menu.**
 - c. **Drag the mouse pointer down to Graph and drag right over the desired attribute name.**

4.6.2.1 Changing a Graph Display

Follow the steps below to change the display of the graph:

1. **Move the mouse pointer over the displayed graph.**
2. **Press MENU and drag the mouse pointer down to Properties and release.**

3. Modify the Properties window of the graph.

To change the viewing angle of the graph:

1. Move the mouse pointer over the displayed graph.
2. Press MENU and drag the mouse pointer down to Controls and release.
3. In the Rotation Angles window, adjust the elevation and rotation of the graph by dragging the mouse pointer on the slider buttons.

To zoom into a portion of the graph:

1. Move the mouse pointer to a point in the graph along the X-axis and press SELECT.
2. Drag the mouse pointer to another point along the X-axis.

To merge multiple graphs:

1. In the Results Grapher main window, click SELECT on the names of the graphs you want to merge.
2. Click SELECT on the Merge button.

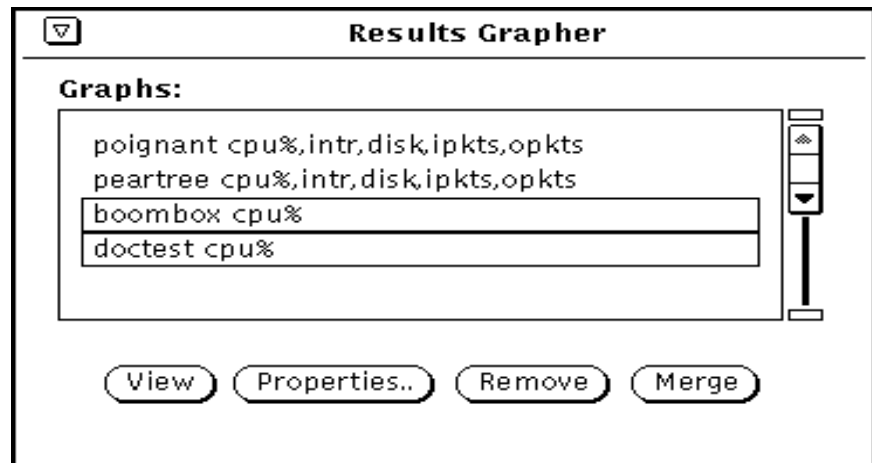


Figure 4-21 Results Grapher Window

4.7 Printing a Graph

You can print graphs (or strip charts) using the OpenWindows Snapshot utility. Snapshot is available from the Tools menu of the Console.

1. **Start the Snapshot utility from the Console Tools menu or from the OpenWindows Programs menu.**
 - In the Console, move the mouse pointer to the Tools button and press MENU to open the Tools menu. Drag the mouse pointer down to Snapshot and release.
 - In an open area of your workstation display, press MENU to open the Workspace menu. Drag the mouse pointer right on Programs, then drag down to the Snapshot option and release.
 - For more information on the Snapshot utility, refer to the *SunOS 4.x DeskSet Environment Reference Guide*.
2. **In the Snapshot window, click SELECT on Window in the Snap Type field, then click SELECT on the Snap button. (See Figure 4-22.)**

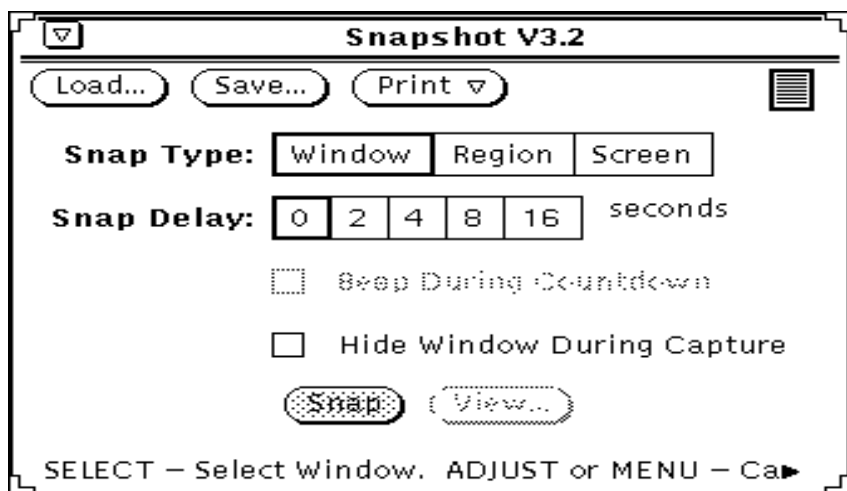


Figure 4-22 Tools—Snapshot Window

3. **Move the mouse pointer to the graph to be printed and click SELECT.** Within a few seconds you should see the message “Snap succeeded” at the bottom of the Snapshot window.

4. In the Snapshot window, press MENU on the Print abbreviated menu button.
5. Drag the mouse pointer down to Print Snap to print the snapshot. Drag the mouse pointer down to Options to set up printer options.

4.7.0.1 *Useful Notes About Displaying and Printing Graphs*

The Results Browser and Grapher run as separate processes and remain running when you quit or close the Console.

By default, graphs are displayed on a black background. To display graphs on a white background, invoke the Grapher from the command line with the `-b` option. You can also modify the Tools menu to add the `-b` option when invoking the Grapher. See Chapter 10, “Customizing SunNet Manager” for more information.

Graphs can be printed by using the OpenWindows Snapshot utility. Snapshot is available from the Tools menu of the Console. See “Printing a Graph” later in this Section for more information.

If the Grapher displays the graphs on a black background, the graphs will print on a black background. To print graphs on a white background, invoke the Grapher with the `-b` option. You can also modify the Tools menu to add the `-b` option when invoking the Grapher. See Chapter 10, “Customizing SunNet Manager” for more information.

4.8 *Viewing and Managing Requests*

The SunNet Manager Console Requests►Summary window allows you to view the status of all requests that have been started from the Console session. From the Requests►Summary window you can stop, kill, or restart multiple requests at a time or modify the properties of a request.

Each request is associated with a request glyph that resides in the subview of the target element. From the Glyph menu for the request you can modify, stop, or kill the request.

This section describes how to view, modify, stop, and kill requests from the Requests Summary window and from the Glyph menu for the request.

4.8.1 Viewing the Status of Requests

The Console Requests>Summary window allows you to view the status of all active or held Console requests. To receive the Requests>Summary pop-up window:

1. Move the mouse pointer over the Requests button and press MENU to open the Requests menu.
2. Drag the mouse pointer down to Summary and release. See Figure 4-23

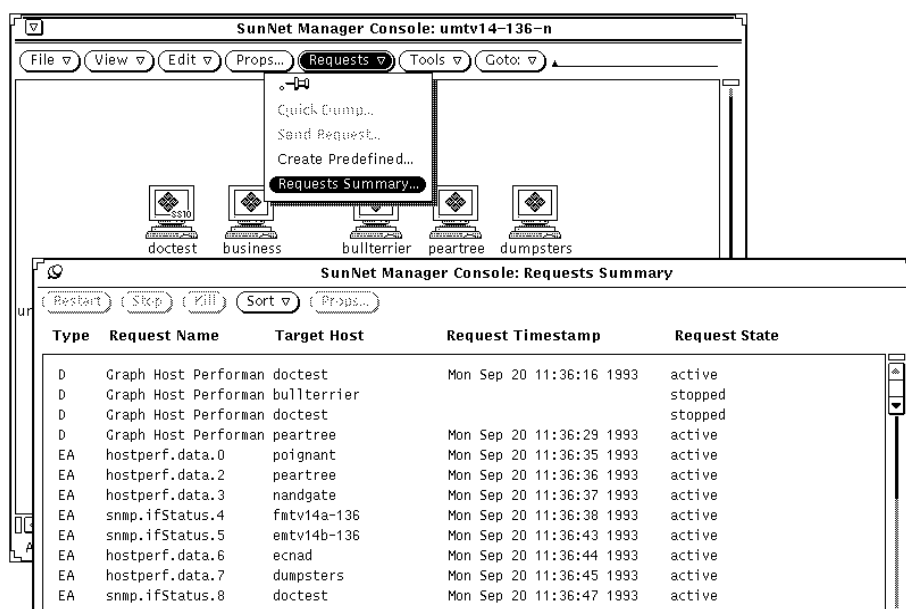


Figure 4-23 Sample Requests—Summary Window

3. To display the requests in a different order:
 - a. Move the mouse pointer over the Sort button and press MENU to open the Sort menu.
 - b. Drag the mouse pointer down and release on the desired sort.

4.9 Request States

Every active or held Console request is shown in one of several *states*:

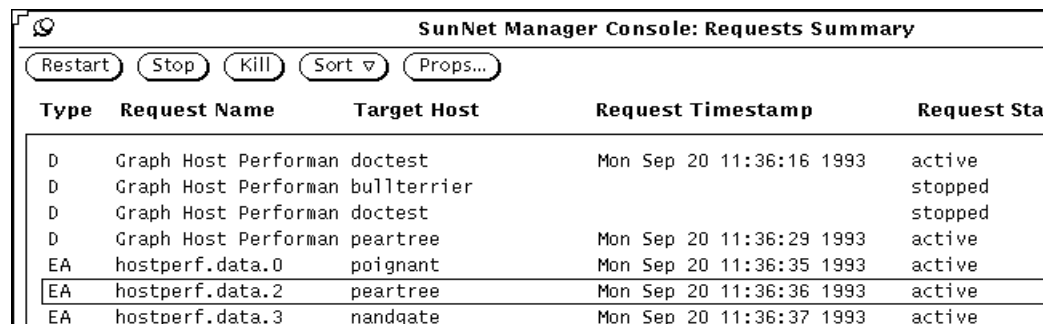
- Not defined
 - Idle
 - Retrying Failed Start
 - Being Started
 - Active
 - Retrying Failed Stop
 - Stopping
 - Scheduled
- When you start a request (you click SELECT on the Start button on the Properties window for the request), the request state is initially *awaiting activation*. Normally, the request state quickly changes to *being activated*, followed by *active*.
 - If an agent error occurs while a request is being activated, the request state changes to *stopped*.
 - An active request that is stopped or killed passes through the states *awaiting stop*, *being stopped*, and *stopped*.
 - A held request (you click SELECT on the Hold button on the Properties window for the request) is shown in *stopped* state.
 - If you specified Delete Request in the Upon Completion field in the Properties window of a request, the request disappears from the Requests>Summary window when the request is completed. If you specified Save Request in the Upon Completion field in the Properties window of a request, the state of the request becomes *stopped* when the request is completed.
 - Note that the states *awaiting activation*, *being activated*, *awaiting stop*, and *being stopped* are normally temporary. If you notice that a request seems to be in one of these states for a long time without changing, you should make sure that the target element is reachable by using the ping agent to do a Quick Dump request.
 - A request that is Scheduled for later is sent to the appropriate agent at the scheduled time.

4.10 Viewing and Modifying Properties of a Request

You can modify a data or event request from the Requests►Summary window or from the request glyph.

To view the properties of a request from the Requests►Summary window:

1. Click **SELECT** on the request whose properties you want to view or modify.
2. Click **SELECT** on the Props button at the top of the window. See Figure 4-24



Type	Request Name	Target Host	Request Timestamp	Request Sta
D	Graph Host Performan	doctest	Mon Sep 20 11:36:16 1993	active
D	Graph Host Performan	bullterrier		stopped
D	Graph Host Performan	doctest		stopped
D	Graph Host Performan	peartree	Mon Sep 20 11:36:29 1993	active
EA	hostperf.data.0	poignant	Mon Sep 20 11:36:35 1993	active
EA	hostperf.data.2	peartree	Mon Sep 20 11:36:36 1993	active
EA	hostperf.data.3	nandgate	Mon Sep 20 11:36:37 1993	active

Figure 4-24 Requests—Summary Window

To view the properties of a request from the request glyph:

1. Move the mouse pointer over the glyph that represents the target element for which you have specified an Event or Data report, and double-click **SELECT**.
Data requests are represented by a meter glyph, while event requests are represented by an alarm clock glyph. When a request is active, its glyph appears black or solid. When a request is stopped or held, its glyph is dimmed.
2. Move the mouse pointer over the glyph that corresponds to the request and press **MENU** to open the Glyph menu for that request.
3. Drag the mouse pointer down to Properties and release. See Figure 4-25.

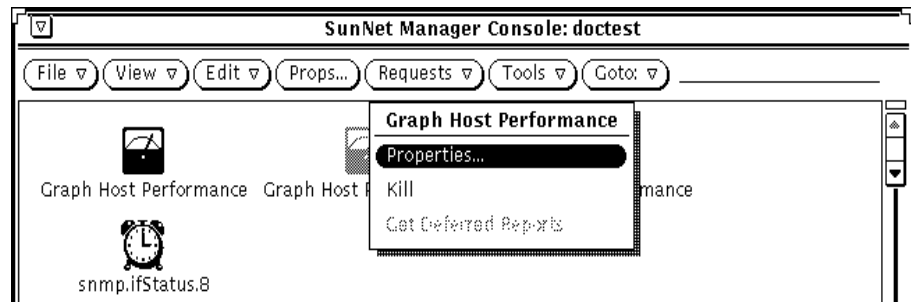


Figure 4-25 Request Glyph Menu

4.10.1 Modifying Report Characteristics or Specifications

Once the Properties window for the request is displayed, you can modify request reporting characteristics or the attribute or event specifications.

To change how an attribute is logged or displayed, select the attribute in the Attribute values scrolling list by pointing the mouse at the entry and clicking SELECT. This causes the current values of the event definition to appear in the fields on the right side of the window. If you modify any of these fields, click the Apply button at the bottom of the window. To delete an attribute from a data request, select the attribute from the Attribute values scrolling list and click the Delete button at the bottom of the window.

4.10.2 Using Data Requests to Monitor Performance

Use data requests to gather comparative data in such areas as network traffic levels or the performance of critical nodes, such as servers and routers. A data request causes an agent to report the values of specified attributes for a target device. You can request a one-time report of current data (a Quick Dump), or reporting of data at periodic intervals. You can see data reports in a real-time graph or log them to a file for later viewing.

Continual data polling can generate significant additional network traffic. You may prefer to collect data only at periodic intervals, to establish an historical record, to help you gauge trends, and to determine normal levels of activity.

Define your own data requests using the Data Requests Properties window or, more conveniently, use one of SunNet Manager's predefined data requests. Table 4-1 lists the predefined data requests shipped with the current version of SunNet Manager.

Table 4-1 SNM Supplied Predefined Data Requests

Request Name	Agent Name	Group Name	Attributes Supported	Function
Graph Host Performance	hostperf	data	cpu% intr disk ipkts opkts	CPU utilization # of device interrupts # of disk transfers # of if input pkts # of if output pkts
Record Disk Space	diskinfo	diskSpace	all	disk space information
Record Host Performance	hostperf	data	all	Statistics of Host
Show Host Interfaces	hostif	if	all	host interface statistics
Show NFS Statistics	rpcnfs	client	all	RPC and NFS statistics
Show Path to Host	ippath	path	all	trace IP packet's path between proxy and target system
Show Routing Statistics	iproutes	routes	all	Routing statistics
Show snmp System Info	snmp	system	all	System information
Show snmp-mibII System Info	snmp-mibII	system	all	System information
Show sun-snmp System Info	sun-snmp	system	all	System information

To view a system's performance in a real-time graph, select the predefined data request "Graph Host Performance." As shown in Figure 4-26, you can launch this data request from the target host's glyph popup menu.

You can launch this request at each critical node. For each such request, you receive a graph such as the one in Figure 4-27.

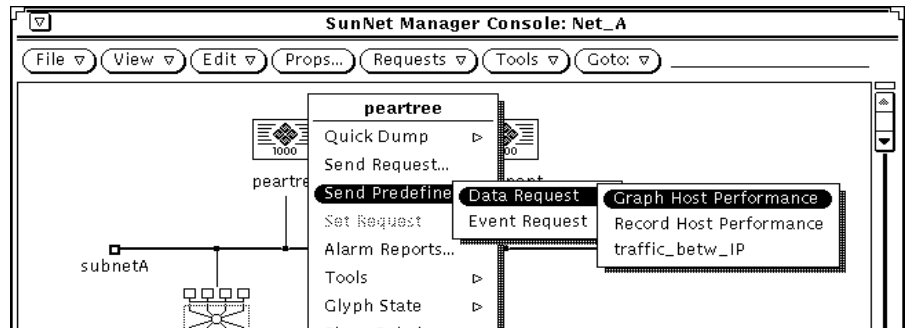


Figure 4-26 Predefined Data Request Using hostperf Agent

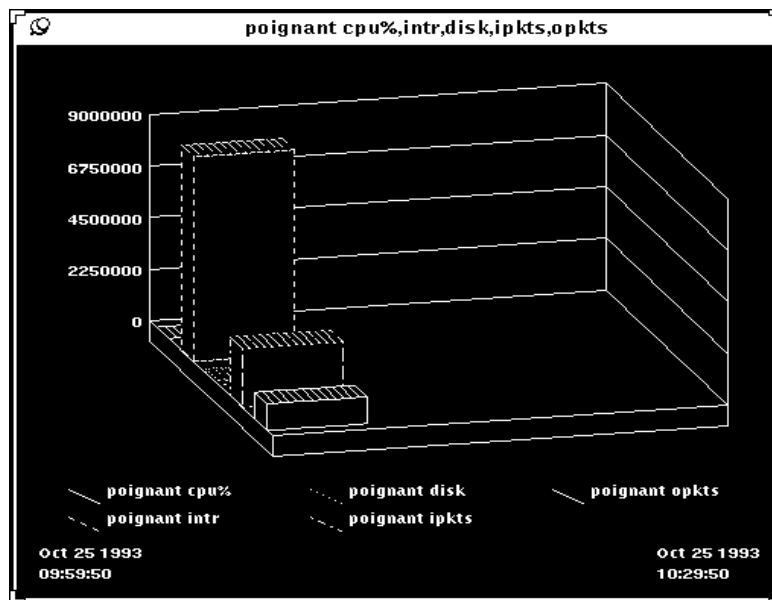


Figure 4-27 Graph from Predefined Request

Example:

You can also use the `hostperf` agent to log data reports to a file. See Figure 4-28. To build such a request for a target device:

1. **Select the device.**
2. **Invoke the Request Builder window through the Console's Requests►Send Request option.**
3. **Select the Data Request type and the `hostperf` agent**
4. **Click on the Apply button to invoke the Data Request Properties sheet.**
5. **To launch the request, click SELECT on the Start button.**

The subview for each device shows the active requests targeted at that device. To access the subview for a device:

6. **Double-click SELECT on the glyph for that device.**
The active requests for that device are represented by glyphs.

SunNet Manager Console: Predefined Data Request

<p>Request Name: <input type="text" value="Record Host Performance"/></p> <p>Agent Schema: <input type="text" value="hostperf"/></p> <p>Group/Table: <input type="text" value="data"/></p> <p>Polling Interval: <input type="text" value="400"/> / <input type="text" value="1"/> seconds</p> <p>Count: <input type="text" value="0"/> / <input type="text" value="1"/> times</p> <p>Key: <input type="text"/></p> <p>Options: <input type="text"/></p> <p>Restart: <input checked="" type="checkbox"/></p> <p>Defer Reports: <input type="checkbox"/></p> <p>On Completion: <input type="checkbox"/> Delete Request</p> <p>Log to File: <input checked="" type="checkbox"/> <input type="text" value="snm-logfiles/network.log"/></p> <p>To Program: <input type="text"/></p> <p>Schedule: <input type="checkbox"/> Off</p> <p>Date Format: <input type="checkbox"/> Month/Day/Year</p> <p>Start Date: <input type="text"/></p> <p>Stop Date: <input type="text"/></p> <p>Start Time: <input type="text"/> / <input type="text"/> AM PM</p> <p>Stop Time: <input type="text"/> / <input type="text"/> AM PM</p>	<p>Selected Attributes:</p> <div style="border: 1px solid black; padding: 5px; min-height: 100px;"> <p>cpu%</p> <p>disk</p> <p>oco ls%</p> <p>oerrs</p> </div> <p style="text-align: right;"> <input type="button" value="Add"/> <input type="button" value="Change"/> <input type="button" value="Delete"/> </p> <p>Available Attributes: <input type="text" value="oerrs"/></p> <p>Data Log: <input type="text" value="True"/></p> <p>Indicator: <input type="text"/></p> <p>Strip Chart: <input type="text" value="None"/></p> <p>Graph Tool: <input type="text" value="None"/></p>
--	--

oerrs added..

Figure 4-28 Data Request Properties Sheet with Log File Specified

4.10.2.1 Copying and Pasting Requests to Track Additional Devices

After a request is launched for one device, you can copy and paste it from the subview of the initial device to the subview of each additional device you want to track. As shown in Figure 4-29, you can use the Console Edit►Copy function to copy requests from the subview of one element to the clipboard. You can then paste this request into the subview of each of multiple elements. This launches a clone of the initial data request for each additional device where you paste the request.

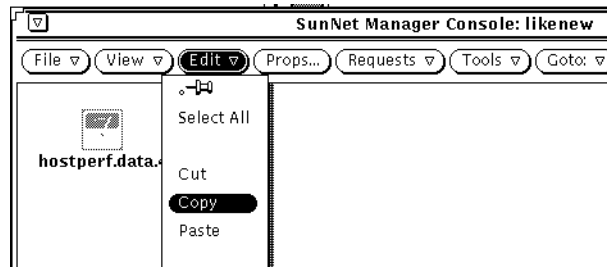


Figure 4-29 Copying a Request from an Element's Subview

To view the data reports from these requests:

1. **Invoke the Results Browser from the Console Tools menu**
2. **Load the log file where the data is stored into the Results Browser through the Browser's File►Load option**

Figure 4-30 shows the Results Browser loaded with the log file specified in the Log to File field of a Data Request Properties Sheet.

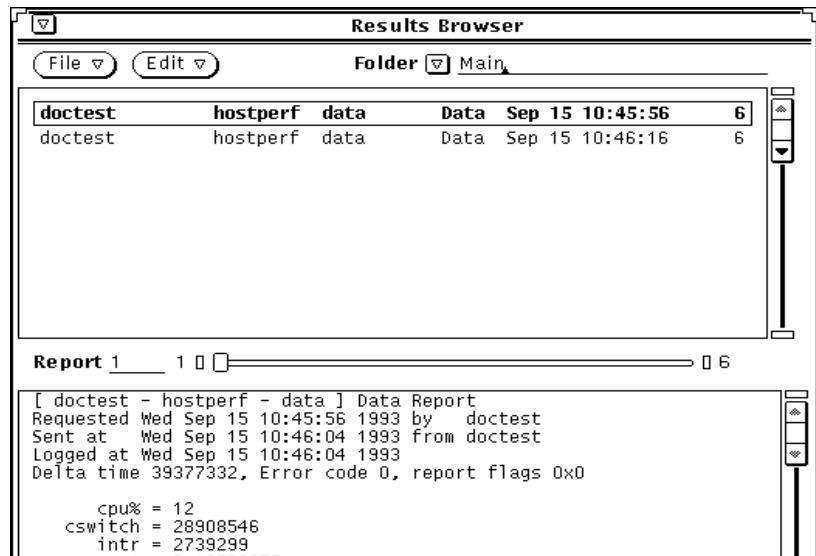


Figure 4-30 Data Report Files in Browser

4.10.3 Example: Assessing Overall Operation of the Network

To quickly assess individual performance characteristics of each critical node, and determine potential problem areas for overall operation of the network, graph the attributes available from the Streams►Graph Grapher menu shown in Figure 4-31. Features such as “zooming” and merging help you refine comparisons of the data you have logged. A detailed discussion of Grapher features is presented in the “Grapher” chapter in “Part 2: Reference.”

To compare data from different streams over a particular attribute, such as `cpu%`, merge the data from the various files into a single graph:

- Invoke the Grapher tool from the Browser.

You receive the window shown in Figure 4-32. The merged graph allows you to quickly compare the CPU utilization of multiple nodes.

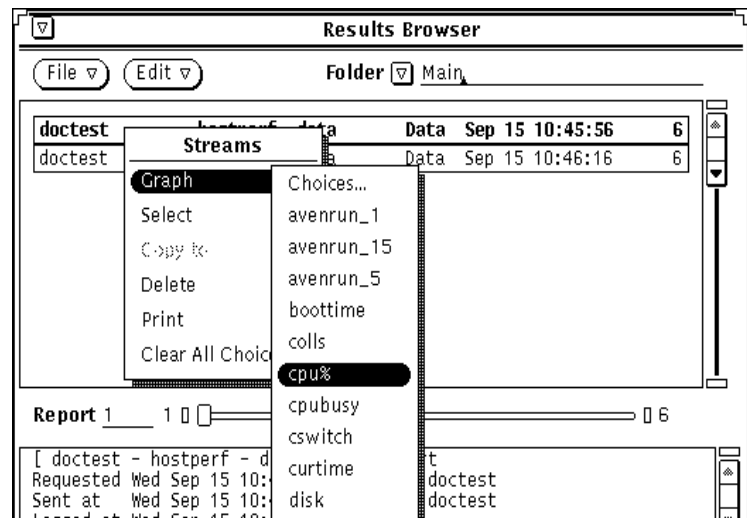


Figure 4-31 Streams►Graph Grapher Menu Data Report Files

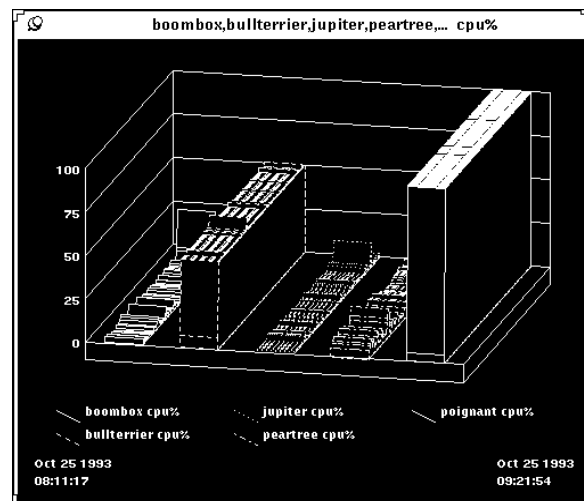


Figure 4-32 Merged Graphs

Specifying Event Requests

This chapter discusses the following topics:

- Specifying an event
- Detecting the presence of an event
- Checking the cause of an event
- Changing the state of a glyph
- Changing the propagation of glyph state changes

An event occurs when specified conditions are met. The Console provides a mechanism that allows you to specify the following characteristics pertaining to an event:

- Condition(s) that define a particular event;
- Visual or audible indications of an event or the action that is to take place when the event occurs;
- Whether or not the visual indications of an event are propagated through the view hierarchy of the Console
- Starting with version 2.3, start time and stop time of the request
- Starting with version 2.3, new requests as part of the event action

When you specify an event, a request is made to the appropriate agent. When an event occurs, the agent sends an event report back to the Console. You can specify various actions for the Console to perform when it receives an event

report. For example, you can have the Console display a visual alarm, play an audio file, send a mail message, send the event report to a program, or any combination of these actions.

Starting with version 2.3, you can stop the request or start another request or perform a combination of both actions.

5.1 *Specifying an Event*

You specify the event conditions, event notification, and actions for individual elements by using the Event Request Properties window.

You can compose your own event request, targeting a specific machine for a specific set of information. More conveniently, you can use a predefined request. These requests save you the trouble of composing requests for individual machines.

The product is shipped with a number of predefined event requests that can accommodate many of your network management needs. You can choose one of these requests, modify it for a specific need, or build your own request.

5.2 *Scheduling Requests Based on Events*

Starting with version 2.3, you can start or stop requests as the result of an event. For example, a ping request may send ICMP packets to a device at regular intervals and generate an event if there is no response. If a device goes down for an extended time and the request continues to launch at the same intervals, a significant number of reports would be generated if no action is taken.

Under such conditions, it would be helpful to stop the event request and put the device in a pending state. This would reduce unnecessary management of network traffic.

With event-based requests, you can:

- Stop the current request
- Start another event request
- Carry out the event request on a different proxy agent using the alternate proxy field

You can combine these options or use them independently.

Use the Properties sheet for an event request to define a schedule (see Figure 5-1).

Screenshot of the SunNet Manager Console: Event Request (emp-lab-dev.snmp.ifStatus) Properties Sheet. The window contains various configuration fields for an event request, including Name, Proxy System, Interval, Count, Key, Restart, Send Once, Defer Reports, On Completion, Options, Alternate Proxy, Schedule, Date Format, Start Date, Stop Date, Start Time, Stop Time, Attributes, Attribute, Relation1, Threshold1, Relation2, Threshold2, Priority, Glyph Effect, Audio Effect, Audio File, Mail To, To Program, Stop Request, and Start Request. At the bottom, there are buttons for Start, Hold, Reset, Apply, and Delete. The key is set to ifIndex and the interface selector is visible.

Figure 5-1 Event Request Properties Sheet

Use this screen together with the Request Builder (as demonstrated in examples that follow) to specify details of how SunNet Manager should handle your request.

5.3 Retrieving Single Attributes

Starting with version 2.3 of the product, you can use an event request to retrieve a single attribute or a subset of attributes. Previously, an event would generate a report containing information about non-requested, as well as requested, attributes in a group.

Here is an example of the type of new report you can generate:

Event Report

```
Mon March 4 16:58:00 1996 [yercaud]: Event: snmp-mibII.system
sysUpTime=26:14:24:68 (Greater Than 00:00:00:00 Priority Low)
```

The report shows the requested attribute, `sysUpTime`. Previously it would also have included non-requested information such as `sysDescr`, `sysObjectID`, `sysContact`, `sysName`, `sysLocation`, and `sysServices`. You can see the type of report previously generated by specifying `false` on the keyword `get-requested-attributes-only` in the `snm.conf` file.

5.3.1 Sending an Event Request Through the Console Requests Menu

1. Click **SELECT** on the glyph that represents the target element.
2. Press **MENU** on the **Requests** button in the control area of the Console and release **MENU** over **Send Request**.

The window shown in Figure 5-2 appears. The Request Action field is automatically set to **Send Request**.

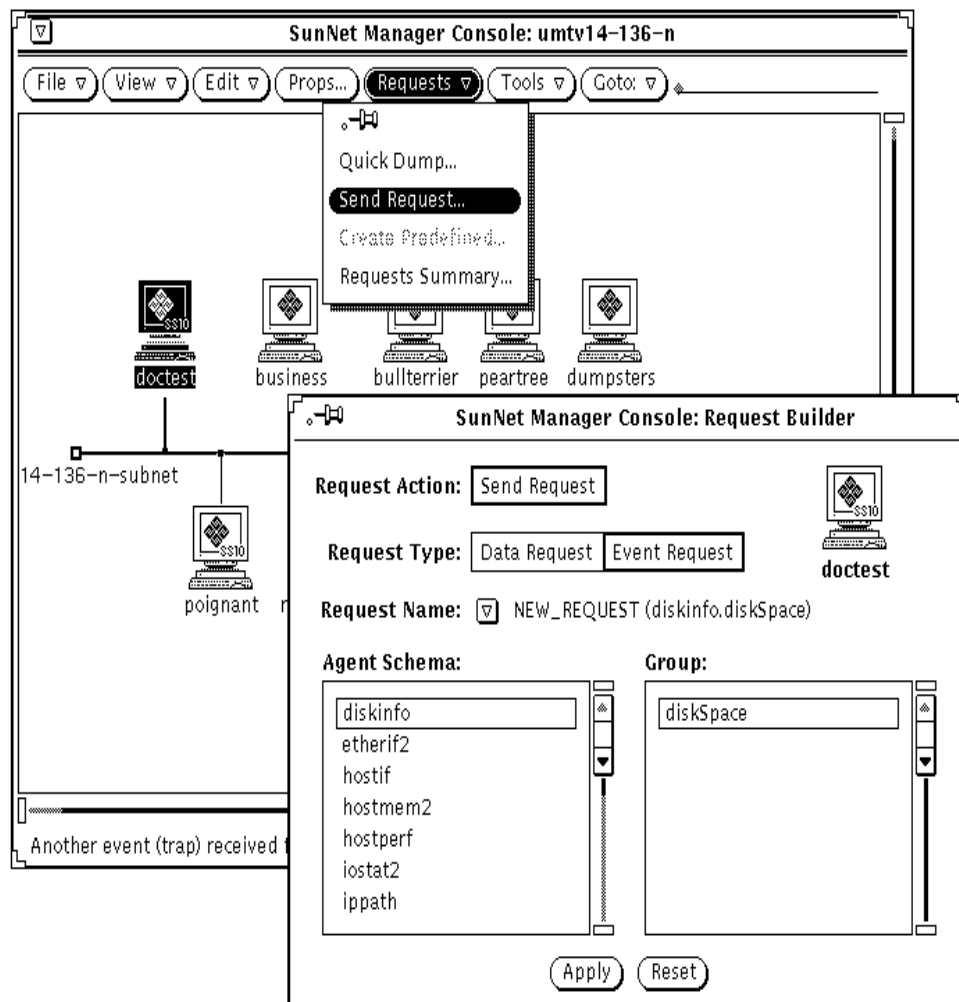


Figure 5-2 Sample Request Builder Window

3. Click **SELECT** on the Request Type you want to send (event, in this example). Press **MENU** on the Request Name abbreviated menu button. You receive the menu shown in Figure 5-3.

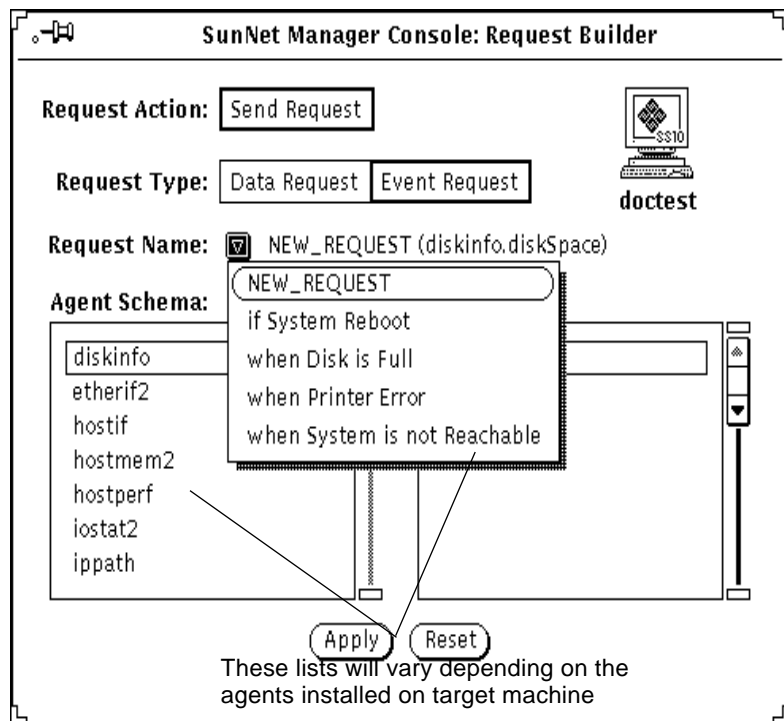


Figure 5-3 Request Name Menu

4. Choose a predefined request from the Request Name menu.
5. If you need to create your own request, do the following:
 - a. Click SELECT on an agent in the Agent Schema scrolling list
 - b. Click SELECT on a group name in the Group scrolling list.
 - c. Choose NEW_REQUEST from the Request Name pull-down menu.
6. Click SELECT on the Apply button.
You receive a Data Request Properties window, as shown in Figure 5-4.
7. Examine the window and make any changes you want.

SunNet Manager Console: Event Request (emp-lab-dev.snmp.ifStatus)

Name: _____

Proxy System: localhost

Interval: 0

Count: 0

Key: _____

Restart:

Send Once:

Defer Reports:

On Completion: Delete Request

Options: _____

Alternate Proxy: _____

Schedule: Off

Date Format: Month/Day/Year

Start Date: _____

Stop Date: _____

Start Time: _____ AM PM

Stop Time: _____ AM PM

Attributes:

Attribute: ifIndex

Relation1: Threshold Not Set

Threshold1: _____

Relation2: Threshold Not Set

Threshold2: _____

Priority: Low

Glyph Effect: Blink Glyph

Audio Effect: None

Audio File: _____

Mail To: _____

To Program: _____

Stop Request:

Start Request: _____

Start Hold Reset Apply Reset Delete

key: ifIndex Interface selector

Figure 5-4 Sample Event Request

1. On the left side of the request window, modify or fill in the report characteristics fields.
2. On the right side of the request window, select the attributes on which you want the agent to send data, and how you want to view the data.
3. Press MENU on the Attribute menu button and release MENU over the attribute you want.
4. Click SELECT on Apply to add the attribute to your request. After selecting and applying attributes and their viewing options, click SELECT on the Start button on the lower left side of the window to send the event request to the agent.

For a description of the fields in the Event Request template, refer to “Part2: Reference.”

5.3.1.1 Using Event Requests to Monitor Critical Nodes

Use the Event Request Properties window to specify conditions that define an event and the type of notification you want to receive. You can compose your own event request or, more conveniently, use a predefined request. Predefined requests included with the current product are summarized in Table 5-1.

Table 5-1 Predefined Event Requests Supplied with SunNet Manager

Request Name	Agent Name	Group Name	Attributes Supported	Event to be Reported
when Disk is Full	diskinfo	diskSpace	capacity	disk file system is full
If System Reboot	hostperf	data	uptime	if system reboots
when Printer Error	lpstat	status	statusCode	line printer error
when System is not Reachable	ping	reach	reachable	system not reachable

The examples that follow show how to send event requests.

5.3.2 Sending Predefined Event Requests through the Glyph Menu

5.3.2.1 Example: Monitoring Node Availability with Ping

To be informed immediately if any of your critical nodes goes down, use an event request with the `ping` agent. For convenience, you can use the predefined event request “when System is not Reachable” and specify the default (blinking glyph).

Here are the steps:

1. **Display the glyph menu by pressing the MENU button over the glyph representing the target device.**

2. Select the predefined event request from this menu as shown in Figure 5-5.

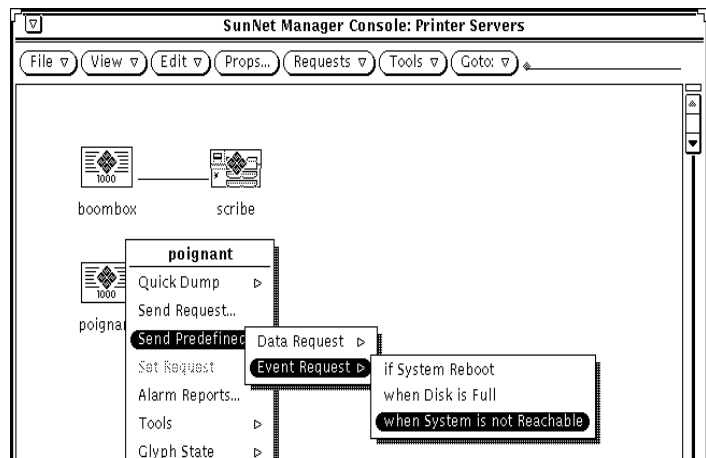


Figure 5-5 Sending Predefined Event Request

Using this method, the predefined request, “when System is not Reachable,” does not take advantage of the “decay” feature. The default is for the glyph to blink when an event occurs, as indicated on the Properties sheet in Figure 5-6.

Screenshot of the SunNet Manager Console: Event Request (emp-lab-dev.snmp.ifStatus) properties sheet. The form is divided into several sections:

- Name:** _____
- Proxy System:** localhost
- Interval:** 0
- Count:** 0
- Key:** _____
- Restart:**
- Send Once:**
- Defer Reports:**
- On Completion:** Delete Request
- Options:** _____
- Alternate Proxy:** _____
- Schedule:** Off
- Date Format:** Month/Day/Year
- Start Date:** _____
- Stop Date:** _____
- Start Time:** _____ AM PM
- Stop Time:** _____ AM PM
- Attributes:** _____
- Attribute:** ifIndex
- Relation1:** Threshold Not Set
- Threshold1:** _____
- Relation2:** Threshold Not Set
- Threshold2:** _____
- Priority:** Low
- Glyph Effect:** Blink Glyph
- Audio Effect:** None
- Audio File:** _____
- Mail To:** _____
- To Program:** _____
- Stop Request:**
- Start Request:** _____

Buttons at the bottom: Start, Hold, Reset, Apply, Reset, Delete.

key: ifIndex Interface selector

Figure 5-6 Sample Event Request Properties Sheet w/Blink Glyph Effect

To change the default, use the Request Builder, as explained below.

Using Request Builder

Use the “decay” feature to cause the device glyph to turn blue (the default color) or to the color you have customized, to alert you when a device has been unreachable at any time since you last cleared an event for it.

Here are the steps:

1. Select the target router and access Request Builder by pulling down the Console Requests menu, as shown in Figure 5-7.

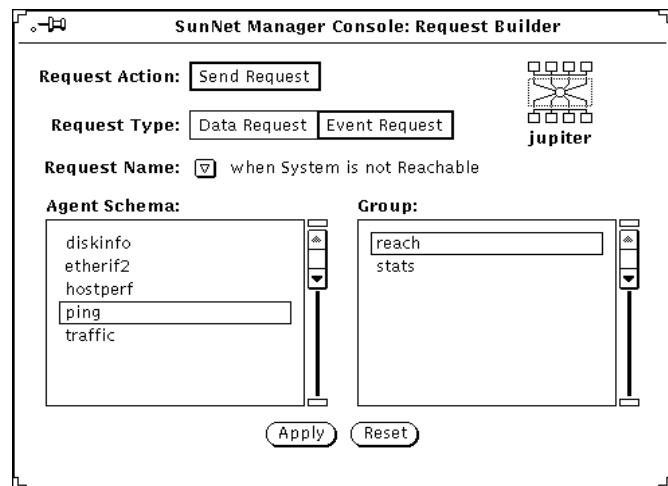


Figure 5-7 Predefined Event Request Builder

2. Select the agent (`ping`), attribute group (`reach`), request type (Event Request), and then “when System is not Reachable” — the name of the predefined event request to use as a starting point for specifying the new event request.
3. Click SELECT on the Apply button to display the Properties sheet.
4. Create a new event request to take advantage of the “decay” feature by:

- a. pressing MENU on the Glyph Effect button and releasing MENU over the Color By Priority option. See Figure 5-8

SunNet Manager Console: Predefined Event Request

<p>Request Name: <input type="text" value="System is not Reachable"/></p> <p>Agent Schema: ping</p> <p>Group/Table: reach</p> <p>Polling Interval: 600 <input type="text"/> second</p> <p>Count: 0 <input type="text"/> times</p> <p>Key: <input type="text"/></p> <p>Options: <input type="text"/></p> <p>Restart: <input checked="" type="checkbox"/></p> <p>Send Once: <input type="checkbox"/></p> <p>Defer Reports: <input type="checkbox"/></p> <p>On Completion: <input type="text" value="Save Request"/></p> <p>Schedule: <input type="text" value="Off"/></p> <p>Date Format: <input type="text" value="Month/Day/Year"/></p> <p>Start Date: <input type="text"/></p> <p>Stop Date: <input type="text"/></p> <p>Start Time: <input type="text"/> <input type="text" value="AM"/> <input type="text" value="PM"/></p> <p>Stop Time: <input type="text"/> <input type="text" value="AM"/> <input type="text" value="PM"/></p>	<p>Selected Attributes:</p> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;"> <input type="text" value="reachable"/> </div> <p style="text-align: right;"> <input type="button" value="Add"/> <input type="button" value="Change"/> <input type="button" value="Delete"/> </p> <p>Available Attributes: <input type="text" value="reachable"/></p> <p>Relation1: <input type="text" value="Not Equal To"/></p> <p>Threshold1: <input type="text" value="true"/></p> <p>Relation2: <input type="text" value="Threshold Not Set"/></p> <p>Threshold2: <input type="text"/></p> <p>Priority: <input type="text" value="Medium"/></p> <p>Glyph Effect: <input checked="" type="text" value="Color by Priority"/></p> <div style="border: 1px solid black; padding: 2px;"> <p>Audio Effect: <input type="text" value="Blink Glyph"/></p> <p>Audio File: <input type="text" value="Dim Glyph"/></p> <p>Mail To: <input type="text" value="Color by Priority"/></p> <p>To Program: <input type="text"/></p> <p>Stop Request: <input type="text" value="None"/></p> <p>Start Request: <input type="text" value="Pending State"/></p> </div>
---	---

Request Name: when System is not Reachable

Figure 5-8 Event Request Properties Sheet w/Color by Priority Glyph Effect

- b. clicking SELECT on Apply to add the attribute to your request
- c. clicking SELECT on the Start button to send the event request to the agent.

5.3.2.2 Example: Using sysUpTime to Monitor SNMP Device

The `ping` agent can only determine if a device is available when it polls the device. However, between polls, you may want to be notified if routers have gone down momentarily and then become available again. Use the SNMP `sysUpTime` attribute to check for this on devices that support Simple Network Management Protocol (SNMP).

1. Build an event request by pressing **MENU** over the target element
2. Select **Send Request** to invoke the Request Builder window, as shown in Figure 5-9.
3. Select **Event Request**, the `snmp-mibII` agent, and the `system` group.

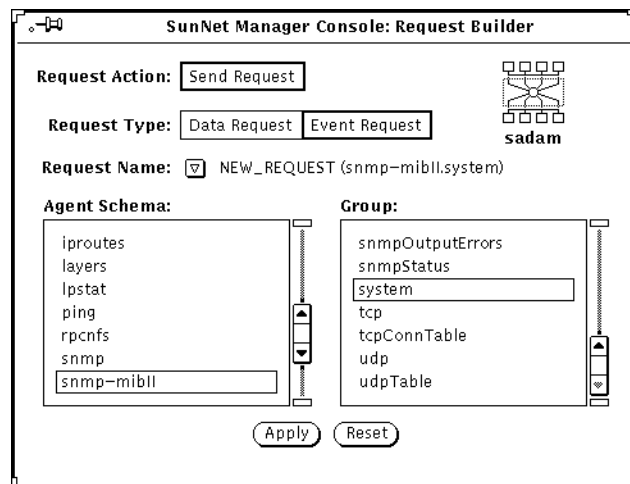


Figure 5-9 Accessing `snmp` `system` Group for Event Request

4. Click **SELECT** on the **Apply** button to access the Event Request Properties sheet, Figure 5-10.

SunNet Manager Console: Event Request (wayout.snmp-mibII.system)

Name: _____

Proxy System: localhost

Interval: 600 [▲▼]

Count: 0 [▲▼]

Key: _____

Restart:

Send Once:

Defer Reports:

On Completion: Save Request

Options: _____

Alternate Proxy: _____

Schedule: Off

Date Format: Month/Day/Year

Start Date: _____

Stop Date: _____

Start Time: _____ [▼] [AM PM]

Stop Time: _____ [▼] [AM PM]

Attributes:

Attribute: sysUpTime

Relation1: Decreased By More Than

Threshold1: 1

Relation2: Threshold Not Set

Threshold2: _____

Priority: High

Glyph Effect: Blink Glyph

Audio Effect: None

Audio File: _____

Mail To: _____

To Program: _____

Stop Request:

Start Request: _____

[Start] [Hold] [Reset] [Apply] [Reset] [Delete]

system Group sysUpTime

Figure 5-10 SysUpTime Event Request Properties Sheet

In Figure 5-10, the following values have been set in the event request fields.

- **Attribute:** sysUpTime
- **Relation:** Decreased by More Than
- **Threshold1:** 1. An event is generated when the value of sysUpTime decreases by more than.001 seconds. If the system is up less time than on the previous poll, the system went down between polls. sysUpTime measures time, in hundredths of a second, since the last system restart.
- **Interval:** 600 seconds (ten minutes)
- **Count:** 0 (to continue indefinitely)
- **On Completion:** Save Request (to prevent the request from being deleted if you stop it).
- **Priority:** High
- **Glyph Effect:** Glyph will blink to indicate a new event.

5. To implement these selections, click **SELECT** on the **Apply** button. The attribute name will then appear in the window at the upper right.
6. Click **SELECT** on the **Start** button to launch the request.

5.3.2.3 Example: Using `ifOperStatus` to Monitor Router Interfaces

Use the SNMP `ifOperStatus` attribute to monitor the availability of router interfaces.

1. Select the target router,
2. To access the Request Builder, select **Send Request** from the glyph popup menu or the **Console Requests** pulldown menu.
3. Select **Event Request**, the `snmp` agent, and the `ifStatus` group, as shown in Figure 5-11.

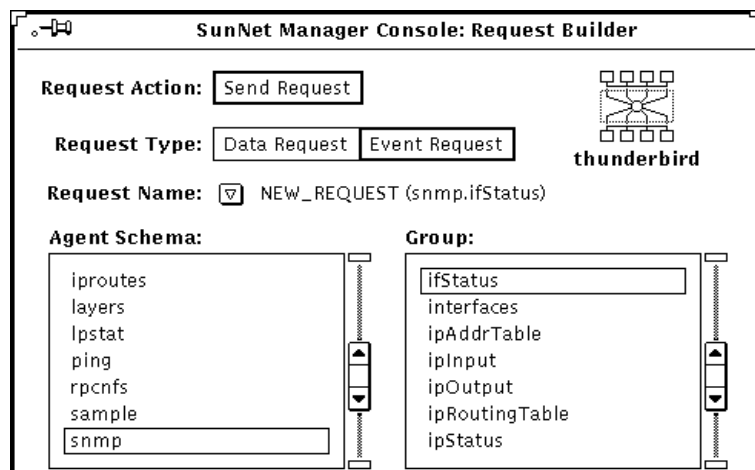


Figure 5-11 Accessing `snmp ifStatus` Group for Event Request

4. Click **SELECT** on the **Apply** button to access the **Event Request Properties** sheet shown in Figure 5-12.
5. Click **SELECT** on **Start** to send the request.

SunNet Manager Console: Event Request (emp-lab-dev.snmp.ifStatus)

Name: _____

Proxy System: localhost

Interval: 0 ▲▼

Count: 0 ▲▼

Key: _____

Restart:

Send Once:

Defer Reports:

On Completion: Delete Request

Options: _____

Alternate Proxy: _____

Schedule: Off

Date Format: Month/Day/Year

Start Date: _____

Stop Date: _____

Start Time: _____ AM PM

Stop Time: _____ AM PM

Attributes:

ifOperStatus

Attribute: ifOperStatus

Relation1: Not Equal To

Threshold1: 1

Relation2: Threshold Not Set

Threshold2: _____

Priority: Low

Glyph Effect: Blink Glyph

Audio Effect: None

Audio File: _____

Mail To: _____

To Program: _____

Stop Request:

Start Request: _____

Attribute ifOperStatus added.

Figure 5-12 Event Request Properties Sheet for ifOperStatus Request

A partial list of values entered in the Event Request Fields are shown below:

Attribute: IfOperStatus

Relation: Not Equal to

Threshold1: 1. The condition that defines a critical event is when the interface is not up. The interface is not up if ifOperStatus has a value other than 1.

Key: If this field is blank, an event is generated if any of the router's interfaces are not up. To test for a particular interface, enter the ifIndex number for that router in the Key field.

5.3.2.4 Generating a Report on Router Interface Status

To generate a one-time report to check the status of all interfaces on a router:

1. Click MENU on the target glyph.

2. Click **SELECT on Quick Dump**►snmp►ifStatus.

This Quick Dump returns the status of all interfaces.

5.3.3 Example: Monitoring Network Traffic Load Using Request Builder

Periodic data reporting can help you determine thresholds in capacity utilization rates or traffic levels where problems occur. You can use these threshold levels to define event requests that tell you when these conditions occur.

In the example below, the `traffic` agent is used to generate an event report when there is an unusually high level of activity between any two nodes on the network. In this case, the particular threshold may have been determined by previous sampling of traffic levels on this network.

This event request is targeted at the Ethernet bus in each of the network's subnets. To generate the request:

1. **Select the Ethernet bus glyph in a subnet view.**
2. **Invoke the Request Builder by selecting the Console's Requests►Send Request option.**

3. Select Event Request type, the traffic agent, and the BetweenIP group, as shown in Figure 5-13.

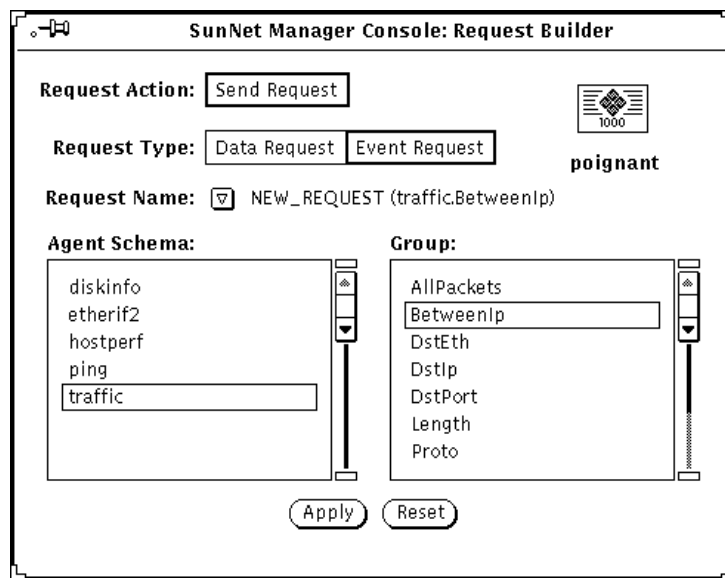


Figure 5-13 Event Request with traffic agent

4. Click **SELECT** on the **Apply** button to invoke the Event Request Properties window as shown in Figure 5-14.

Screenshot of the SunNet Manager Console: Event Request (emp-lab-dev.traffic.BetweenIp) window. The window is divided into two main sections: configuration on the left and attributes on the right. The configuration section includes fields for Name, Proxy System (localhost), Interval (60), Count (0), Key, Restart, Send Once, Defer Reports, On Completion (Delete Request), Options, Alternate Proxy, Schedule (Off), Date Format (Month/Day/Year), Start Date, Stop Date, Start Time, and Stop Time. The attributes section includes Attribute (pkts), Relation1 (Greater Than), Threshold1 (400), Relation2 (Threshold Not Set), Threshold2, Priority (High), Glyph Effect (Color by Priority), Audio Effect (None), Audio File, Mail To (jack.netman@eng.com), To Program, Stop Request, and Start Request. At the bottom, there are buttons for Start, Hold, Reset, Apply, and Delete. A status bar at the bottom indicates 'IP Traffic between two hosts' and 'packets between the two IP addresses'.

Figure 5-14 Example of traffic Event Request

5. Click **SELECT** on the **Start** button to launch this request.

All fields in the Event Request Properties sheet are described in Chapter 15, “Requests Management.”

The fields in this example have been completed as follows:

- **Attribute:** Set to “pkts”.
- **Relation1:** Set to Greater Than.
- **Threshold1:** Set to 400.

In observing the network, you may have noticed that if more than 400 packets are sent between two nodes, it creates an abnormally high rate of network activity. “Greater Than 400,” then, defines the condition for generating an event notification.

- **Glyph Effect:** Set to Color by Priority. This allows you to take advantage of the “decay” feature, to indicate that this condition has occurred at some time since you last cleared an event. You can customize the color of the glyph. You need not use blue.
- **Mail To:** You can specify your e-mail address in this field. Any e-mail message generated by this event would contain the IP addresses of the two nodes whose packet activity exceeded 400.
- **Interval:** This value is set to 60 so the agent polls every minute to compare the result with the specified threshold value.
- **Count:** This value is left at 0 (the default) so that the event request will continue indefinitely.

After launching this request for the gateway node in one subnet, you can copy it from the subview of that node and paste it into the subview of a gateway node on each additional Ethernet subnet you want to monitor. Copying the request to the subview of an element launches a clone of that event request targeted at the node where the request has been copied.

5.3.3.1 *Alternative Ways to View the Properties Sheet for Event Requests*

1. **Double-click on the target glyph.**
2. **Click SELECT on the glyph for the request**
3. **Click SELECT on the Props button in the Console’s control area.**
4. **Invoke Requests►Requests Summary from the Console’s control area.**
5. **In the Requests Summary window, find and click SELECT on your request,**
6. **Click SELECT on the Props button in the Requests Summary window.** In that window, the Select and Sort menus can be very useful in finding your request.

5.3.3.2 Example: Managing Log Files Using Predefined Request

You can use event requests to help you manage the management station, itself. In this example, an event request is defined to generate an alarm when the file system where the Console log files are mounted (/var), is more than 90% full..

1. Click **SELECT** the glyph representing your management station.
2. invoke the Request Builder through the Requests►Send Request option.
3. As shown in Figure 5-15, select the diskinfo agent and Event Request in the Request Builder.

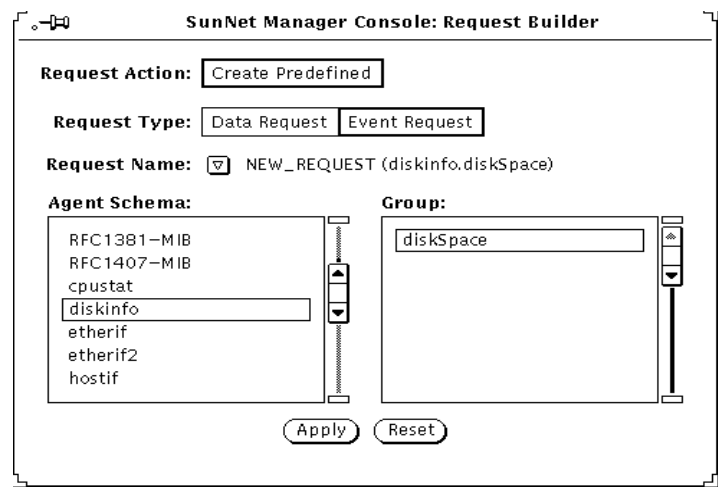


Figure 5-15 Managing Log Files: Request Builder Selection

4. Click **SELECT** the “Apply” button, to get the Event Request Properties sheet in Figure 5-16.

SunNet Manager Console: Predefined Event Request

<p>Request Name: <input type="text" value="check_snm_logfiles"/></p> <p>Agent Schema: <input type="text" value="diskinfo"/></p> <p>Group/Table: <input type="text" value="diskSpace"/></p> <p>Polling Interval: <input type="text" value="600"/> <input type="button" value="▲"/> <input type="button" value="▼"/> second</p> <p>Count: <input type="text" value="0"/> <input type="button" value="▲"/> <input type="button" value="▼"/> times</p> <p>Key: <input type="text" value="/var"/></p> <p>Options:</p> <p>Restart: <input checked="" type="checkbox"/></p> <p>Send Once: <input type="checkbox"/></p> <p>Defer Reports: <input type="checkbox"/></p> <p>On Completion: <input type="button" value="▼"/> Save Request</p> <p>Schedule: <input type="button" value="▼"/> Off</p> <p>Date Format: <input type="button" value="▼"/> Month/Day/Year</p> <p>Start Date: <input type="text" value="06/30/96"/></p> <p>Stop Date: <input type="text" value="06/30/96"/></p> <p>Start Time: <input type="text" value="3:00"/> <input type="button" value="▼"/> <input type="button" value="AM"/> <input type="button" value="PM"/></p> <p>Stop Time: <input type="text" value="3:15"/> <input type="button" value="▼"/> <input type="button" value="AM"/> <input type="button" value="PM"/></p>	<p>Selected Attributes:</p> <div style="border: 1px solid black; padding: 5px; min-height: 50px;"> <input type="text" value="capacity"/> </div> <p style="text-align: right;"><input type="button" value="Add"/> <input type="button" value="Change"/> <input type="button" value="Delete"/></p> <p>Available Attributes: <input checked="" type="checkbox"/> capacity</p> <p>Relation1: <input checked="" type="checkbox"/> Greater Than</p> <p>Threshold1: <input type="text" value="90"/></p> <p>Relation2: <input checked="" type="checkbox"/> Threshold Not Set</p> <p>Threshold2: <input type="text"/></p> <p>Priority: <input checked="" type="checkbox"/> Low</p> <p>Glyph Effect: <input checked="" type="checkbox"/> Blink Glyph</p> <p>Audio Effect: <input checked="" type="checkbox"/> None</p> <p>Audio File: <input type="text"/></p> <p>Mail To: <input type="text"/></p> <p>To Program: <input type="text"/></p> <p>Stop Request: <input type="checkbox"/></p> <p>Start Request: <input type="text"/></p> <p style="text-align: center;"><input type="button" value="Define"/> <input type="button" value="Reset"/> <input type="button" value="Undefine"/></p>
--	--

'capacity' added

Figure 5-16 Managing SunNet Manager Log Files: Defining the Event Request

5. Enter the appropriate data in each field.

6. Click SELECT on Define to launch the request.

Each field on the Properties sheet is described in Chapter 15, "Requests Management."

Selected fields in the example above are defined as follows:

Interval: specifies a polling interval of 15 minutes.

Count: 0. The request will continue until an operator stops it.

For event reports, the Interval and Count fields simply specify when the agent is to send a report. A report is forwarded to the Console system only if it has been determined that an event has occurred. For agents that are shipped with SunNet Manager, the value reported for an attribute is the value noted at the reporting interval. Thus, it is possible for event conditions to occur *between* reporting intervals which will not generate an event report.

If a count is specified and a stop date and time are specified, the earlier of the two takes precedence. For example, if a count of 100 is specified and a stop date of 9/13 is specified and a stop time of 12:30 am is specified, the request will stop if 100 occurs before the stop date and time. If the stop date and time occur before the count is reached, the request will stop at the specified stop date and time.

Send Once: If the Send Once field is off (no check mark appears in the accompanying box), an event report will be sent for every event occurrence. If the Send Once field is on (a check mark appears in the accompanying box), it means that after the first event report is received by the Console, the request is killed. To specify Send Once, click SELECT on the box to display a check mark.

Key: /var--in this example. For the diskinfo agent, Key specifies the relevant disk partition -- for example, the file system that would be indicated in the "Mounted On" column in a `df` command output. In this example, the log files are situated in a separate partition.

On Completion: Save Request. The request will not be deleted if we stop it.

Attribute: in this example "capacity" designates the percentage of total capacity used. The condition defining the event is capacity Greater Than 90.

Glyph Effect: Color by Priority. If the event condition is met, the glyph, will turn red (or, starting in version 2.3, to a color you select).

By default, an event's glyph effect options remain in effect for 10 seconds beyond the reporting interval. If another event does *not* occur within this time, the Console cancels the glyph effect. You can increase the 10-second glyph effect overlap time in the Console Properties window. See the description of the Event Effect Overlap Time setting in "Part 2: Reference."

Start Date: 06/30/96. Start the request on this date.

Stop Date: 06/30/96. Do not run the request beyond this date.

Start time: 3:00 p.m. Start the request at this time.

Stop time: 3:15 p.m. Do not run the request beyond this time.

When you specify a start date and time and click Start, the request status is listed as "scheduled" in the request summary window.

5.3.3.3 Available Predefined Requests

Predefined requests available (and the agents listed in the Agent Schema list in a Request Builder window) for a given machine depend on the agents that have been checked off as present in the Properties window for that machine. A machine which does not have SunNet Manager on it, that the IP Discover tool adds to your database, has the `hostperf` and `ping` agents checked off. Information is available from a machine without agents through SunNet Manager's proxy feature, just as if the `hostperf` and `ping` agents were on that machine. A machine with agents installed that the IP Discover tool adds to your database, has the agents checked off that are actually present.

5.3.3.4 Automatically Opening an Icon

You can choose to have a Console window that is closed to an icon open automatically when an event or trap is received. You specify this in the Console's Properties window, available by clicking SELECT in the Props button in the Console's control area. See the description of the Open on Event or Trap setting in "Part2: Reference."

5.4 Detecting the Presence of an Event

This section describes how you can use the Alarm Reports or Event/Trap Reports windows to monitor events. You obtain these from the View menu in the Console's control area (View►Alarm Reports or View►Event/Trap Reports). You can use either of these windows for all elements in a view or for a specific element. An alternative, and perhaps easier, way to detect an event for a specific element is to observe the visual effect of an event as it shows in a glyph.

If an event occurs, a glyph can blink, dim, change color, or remain unchanged. You choose the effect you want in the Properties window at the time you make an event request. SunNet Manager has a "decay" feature that allows you to detect that a condition that caused an event report occurred and that the condition no longer exists. For example, a file server might exceed a threshold you have set for a certain number of NFS transactions, then fall below that threshold. To take advantage of the "decay" feature, you must specify Color by Priority as your Glyph Effect.

See the example, “Sending Predefined Event Requests through the Glyph Menu” on page 5-8 for instructions on specifying a request to take advantage of the “decay” feature.

5.5 *Checking the Cause of an Event*

When an event occurs, a report is returned to the Console that indicates the cause of the event. The Console gives you two ways to view an event report:

- The Event/Traps Reports window, described in this section.
- The Alarms Summary window (available in the View▶Alarm Reports window). This window displays error, event, and trap reports and is described in the Section on “Viewing Alarm Reports” later in this Manual.

To check the cause of an event in the Event/Trap Reports window, perform the following steps:

1. **Press MENU over the View button, and Release MENU over Event/Trap Reports. (See Figure 5-17.)**

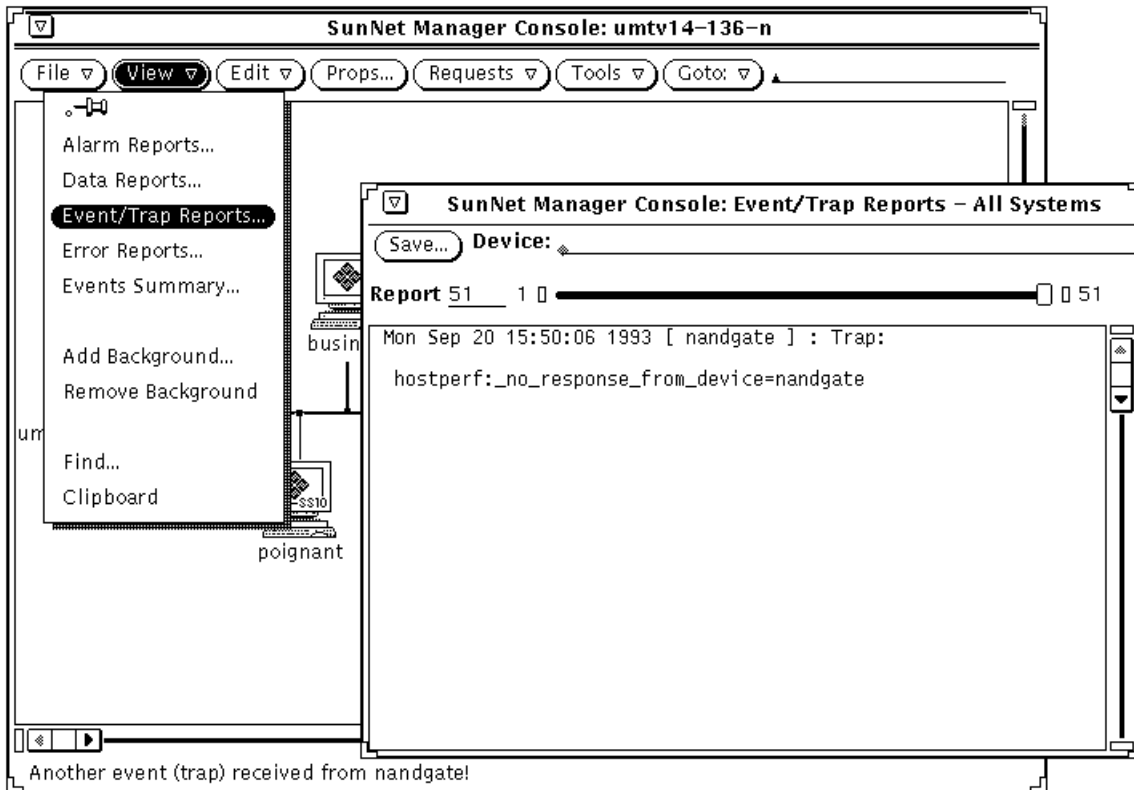


Figure 5-17 View—Event/Traps Window

2. **To examine event and trap reports for a particular system, type in the device name after the Device prompt and press Return.**
To see entries for all elements again, press Ctrl-U to clear the Device line and press Return.

An alternative way to obtain an element-specific event or trap report is to click SELECT on the glyph in the Console window, then invoke View►Event/Trap Reports.

3. **To save the event reports to a file, click SELECT on the Save button.**
A pop-up Save window appears, where you can enter the path and file name where the Event/Trap Reports are to be stored.

4. To browse through entries in the log:

- Press SELECT on the slider and drag the slider to the left or right.
- Click SELECT to the left or right of the slider.
- Click SELECT on either end of the slider bar.
- Enter the number of a report on the Report line and press Return.

5.5.1 Event/Trap Reports Window

Event/Trap Reports entries are time-stamped and show values for all attributes in the specified agent group. The attribute and value that caused the event are flagged with the relational operator and threshold values shown in parentheses to the right.

By default, a maximum of 1000 event reports are displayed in the Event/Trap Reports window. You can change the maximum number in the Console Properties window. To view or change this setting (or any other event- or trap-related setting), click SELECT on Props in the Console's control area, then select Miscellaneous in the Category pulldown menu. See the description of the Maximum Event Reports setting in "Part 2: Reference."

5.6 Changing a Glyph State Back to Normal

A glyph may change its state when an event or a trap is received for the element that is represented by the glyph. For example, if you have specified an event for an element and the notification is "Blink Glyph", the element's glyph state changes from "normal" to "blinking" when the event occurs. Once you have been alerted to the event, you may want to change the state of a glyph back to "normal." This is done using the Glyph State option of the Glyph menu for the element.

5.7 Glyph Pending State

Starting with version 2.3 of SunNet Manager, you can also put a glyph to pending state. You might want to use pending state when a device has been down for an extended period and events continue to be logged against it, or when you know that someone is repairing the device and you do not want new events or traps against the device to have any effect.

When you click Pending On, the glyph goes to pending state, and the color of the object is dimmed. All outstanding events and traps on the device are cleared and the object is “frozen.” New events and traps will not change the appearance of the glyph state or propagate the effect of a trap or event to the parent object. If a parent object is in pending state, a change in the state of the children will not affect the parent.

When you click Pending Off, pending state is turned off, and the object is an end node or device, its glyph state is recalculated based on any outstanding alarms. If there are none, the glyph state is set to normal. If the glyph object is a parent inheriting from its children, its state is recalculated appropriately.

When a pending device is moved to normal state, all outstanding alarms are cleared. If the glyph object is a parent, the state of the children are set to normal, also.

To change the state of a glyph, follow the steps below:

- 1. Press MENU over the glyph whose state you want to change.**
- 2. Release MENU over Glyph State and drag right to the desired glyph state. See Figure 5-18.**

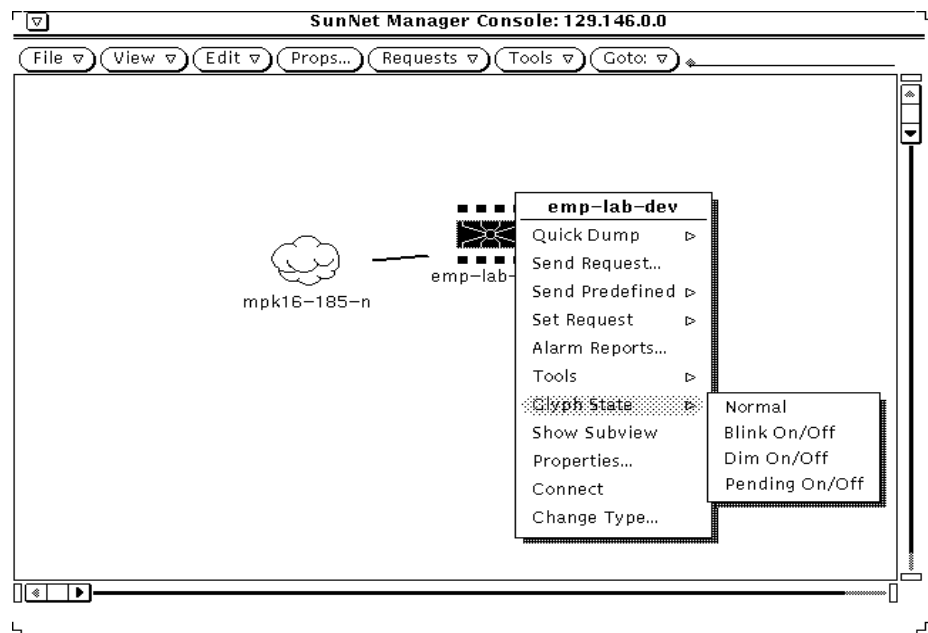


Figure 5-18 Glyph—Glyph State Menu

5.8 Propagation of Glyph States

Whenever a glyph state is changed as a result of an event or a trap, the glyph state is propagated through the view hierarchy. If a glyph is contained in more than one view, its state is propagated through its multiple view hierarchies. When the Console is closed to an icon, glyph state changes reflected in the Home view cause the Console icon to display question marks. These effects are not true for pending state.

When you change the state of a glyph, the new glyph state is propagated as though an event had occurred to change the glyph state. For example, if you change the glyph state of an element from “normal” to “blinking,” the glyph state “blinking” propagates as if an event had occurred.

If you have multiple glyph states propagating into a single view and you change the glyph state for the view to “normal,” the glyph states of the elements contained in the view are also reset to “normal.” For example, if you

change the glyph state of a view from “blinking” to “normal,” this clears the effects of the event that caused the glyph state to change to “blinking.” This allows you to clear multiple events by resetting the glyph state of a view.

You can also choose to have errors that are received from agents cause a change in the glyph state of an element. By default, this is not enabled. See the description of the Errors category of the Console Properties window in “Part 2: Reference.”

5.9 Changing the Propagation of Glyph State Changes

By default, glyph state changes caused by events or traps are propagated through the view hierarchy, except for glyphs in pending state. You can disable glyph state propagation in several ways.

To disable glyph state propagation for a particular view:

1. Press MENU over the glyph that represents the view.
2. Release MENU over Properties to open the Properties window for the view. (See Figure 5-19.)

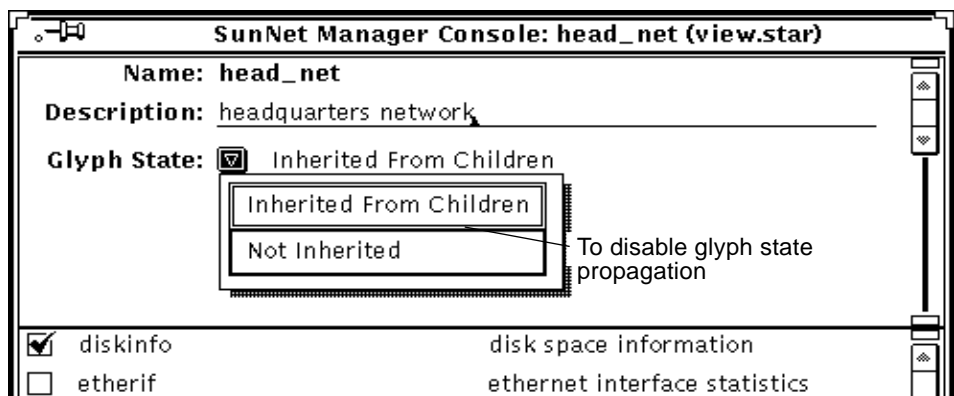


Figure 5-19 Properties Window for a View

3. In the Properties window, press MENU over the Glyph State abbreviated menu button in the top portion of the window.
4. Release MENU over the Not Inherited option.

5. Click **SELECT** on the **Apply** button. To disable glyph state propagation for all views in the database:
 1. Click **SELECT** on the **Props** button in the **Console** window.
 2. In the **Console Properties** window, press **MENU** over the **Category** abbreviated menu button to open the categories menu.
 3. Release **MENU** over **Events and Traps**. (See Figure 5-20.)

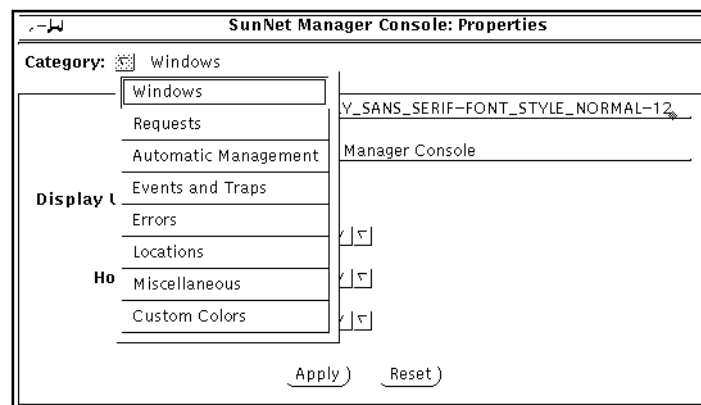


Figure 5-20 Console—Properties—Category Menu

4. In the **Events and Traps** categories window, click **SELECT** in the box next to the **Propagate Event Effect** setting to toggle the check mark off. (See the Figure below.)

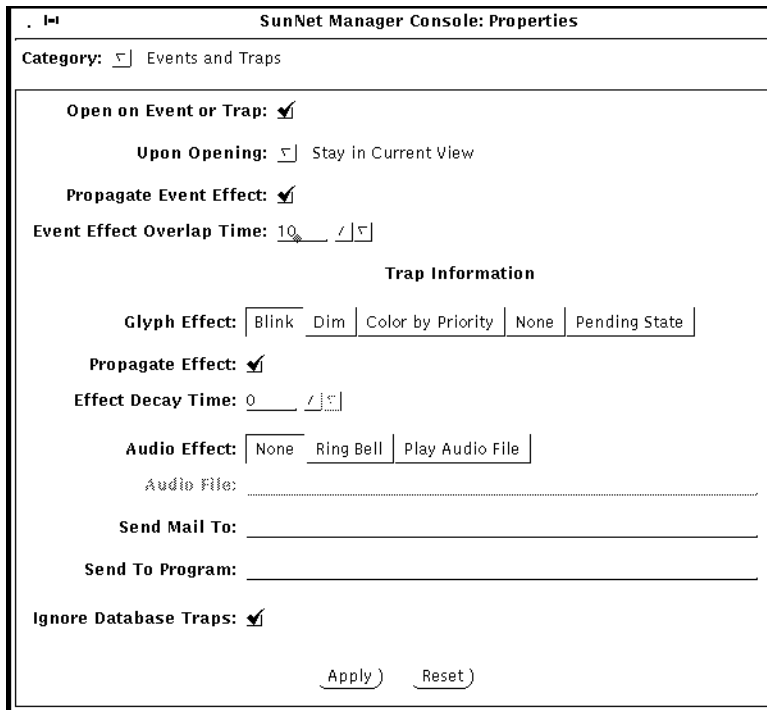


Figure 5-21 Properties—Event/Trap Window

5. Click SELECT on the Apply button.

To disable propagation of glyph states caused by traps:

- 1. Open the Events and Traps category settings in the Console Properties window, as described in the previous steps.**
- 2. Click SELECT in the box next to the Traps Propagate Effect setting to toggle the check mark off.**
- 3. Click SELECT on the Apply button.**

Viewing Reports



This chapter discusses:

- Viewing alarm reports
- Viewing error messages and error reports
- Viewing traps
- Viewing event reports

SunNet Manager returns four types of reports:

- Data report
- Event report
- Error report
- Trap report

The Console groups the last three under the heading of “alarms.” You can view a summary of alarm reports for all elements within a given view. To do this, follow these steps:

- 1. Make sure there are no glyphs selected in the Console window.**
- 2. Press MENU on the View button in the Console window and release MENU over Alarms.**

You receive a window such as the one shown in Figure 6-1.

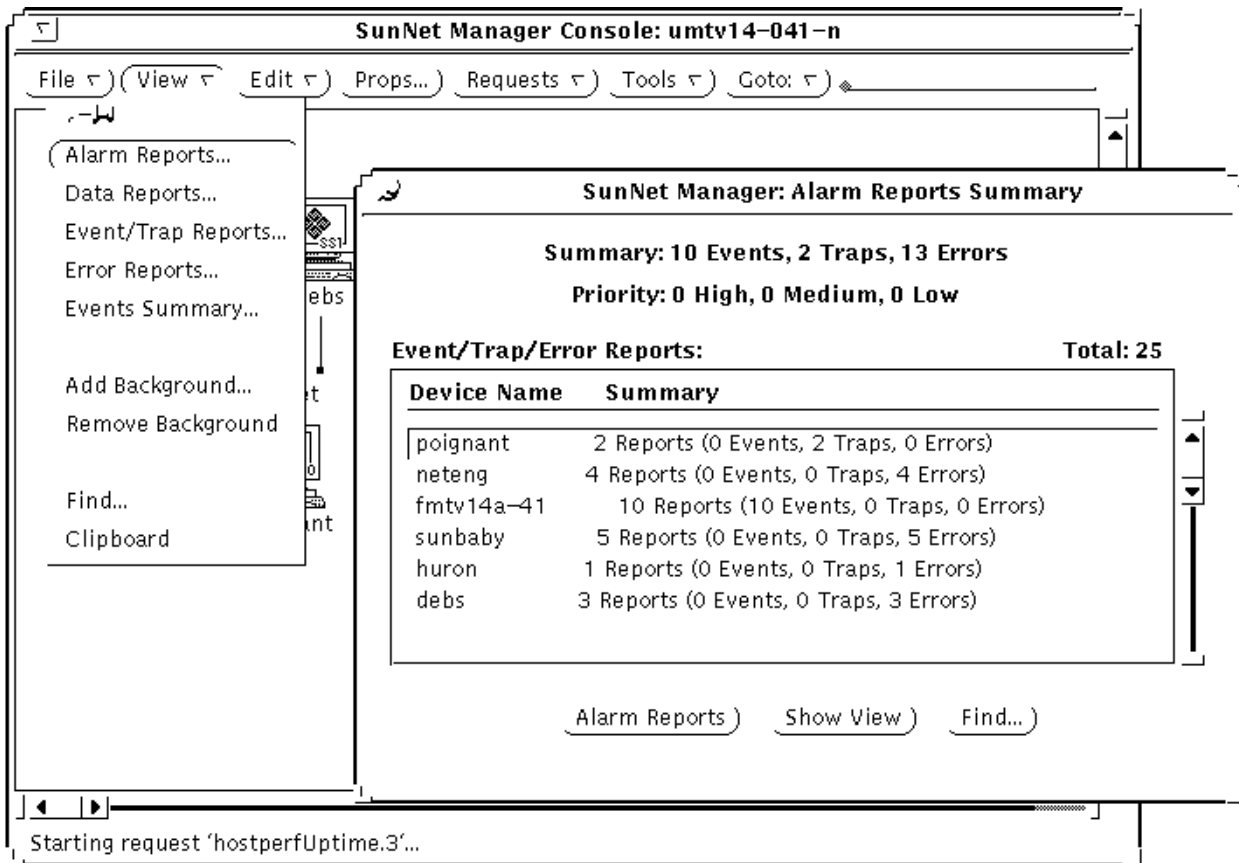


Figure 6-1 Alarm Reports Summary Window

Note – If both the Console Alarm Report Summary window and a device-specific Alarm Summary window are displayed simultaneously and reports continue to arrive for the device, the “number-of-reports-received” totals for the two windows will differ until the Console Alarm Reports Summary window is next updated (every 30 seconds).

The following subsections describe how to use functions of the Alarm Reports feature, including (but restricted to) functions available through the buttons—Alarm Reports, Show View, and Find—at the bottom of the Alarm Reports Summary window.

6.0.1 Finding a Device

To find a device that is not visible in the window's scrolling list, you can scroll through the list or, more conveniently:

1. Click **SELECT** on the **Find** button.
You receive the window shown in Figure 6-2.

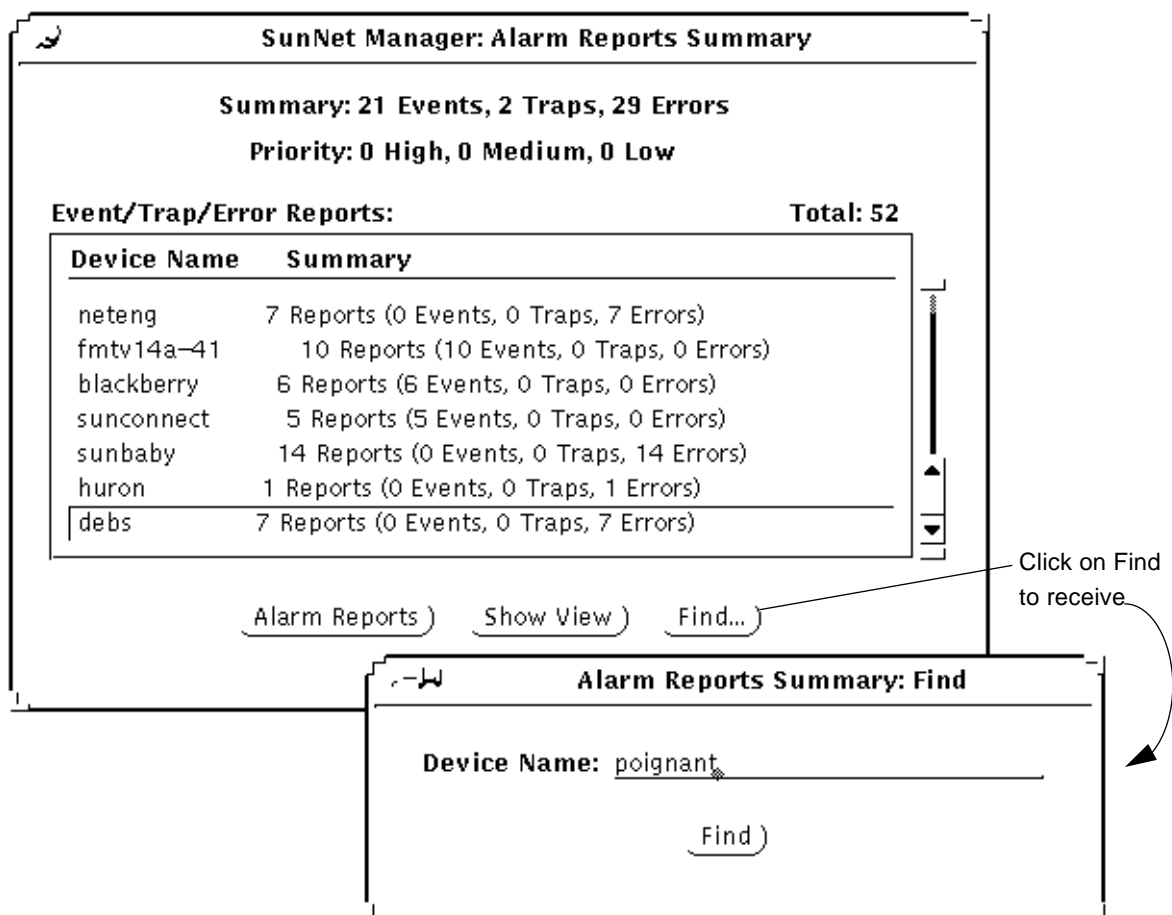


Figure 6-2 Alarm Reports Find Window

2. In the **Device Name** field, enter the name of the network element of the report for which you are looking.

3. Click SELECT on Find.

If a line for a device is present, that line is displayed and highlighted in the Alarm Reports window’s scrolling list. If a line for a device is not present, you receive the message “Device name not found” in a popup window.

6.0.2 Switching Views

Through the Alarm Reports Summary window, you can switch views for a specific device. (This function is also available through the View►Find option in the Console window.) To show the current view for a device and to be able to switch views, do the following:

- 1. Select the specific device in the Alarm Reports Summary window’s scrolling list.**
- 2. Click SELECT on the Show View button.**

You receive a window such as the one shown below.

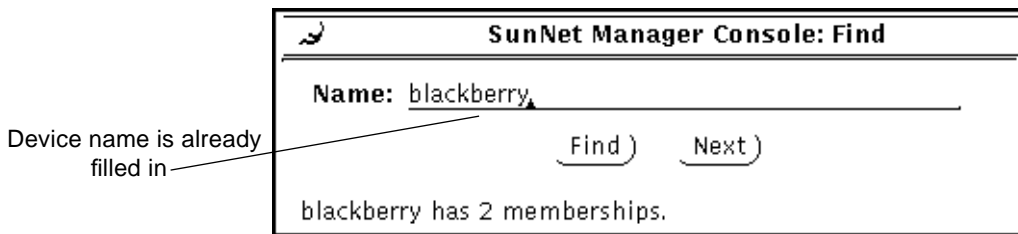


Figure 6-3 Alarm Reports: Show View Window

If a device is in only one view, the Next button in the window above is greyed-out.

- 3. If a device is in multiple views, click SELECT on Next to switch from the current view (as displayed in the Console window) to the next view in which the specified device appears.**

In this context, “next” means the next view on the list of views as displayed by the Console’s Goto menu.

6.0.3 *Obtaining Device-specific Alarm Reports*

You can view further detail for a specific device. To obtain a device-specific report, use any of the following methods:

- Double click on the line for a device in the scrolling list in the Alarm Reports Summary window.
- Click SELECT on the line for a device in the scrolling list in the Alarm Reports window, then click SELECT on the Alarm Reports button at the bottom of the window.
- Click SELECT on the glyph for a device and invoke View►Alarm Reports in the Console menu
- Invoke Alarm Reports from an element's glyph menu.

Using any of the preceding methods, you then receive an Alarm Reports window for the specified device, as shown in Figure 6-4.

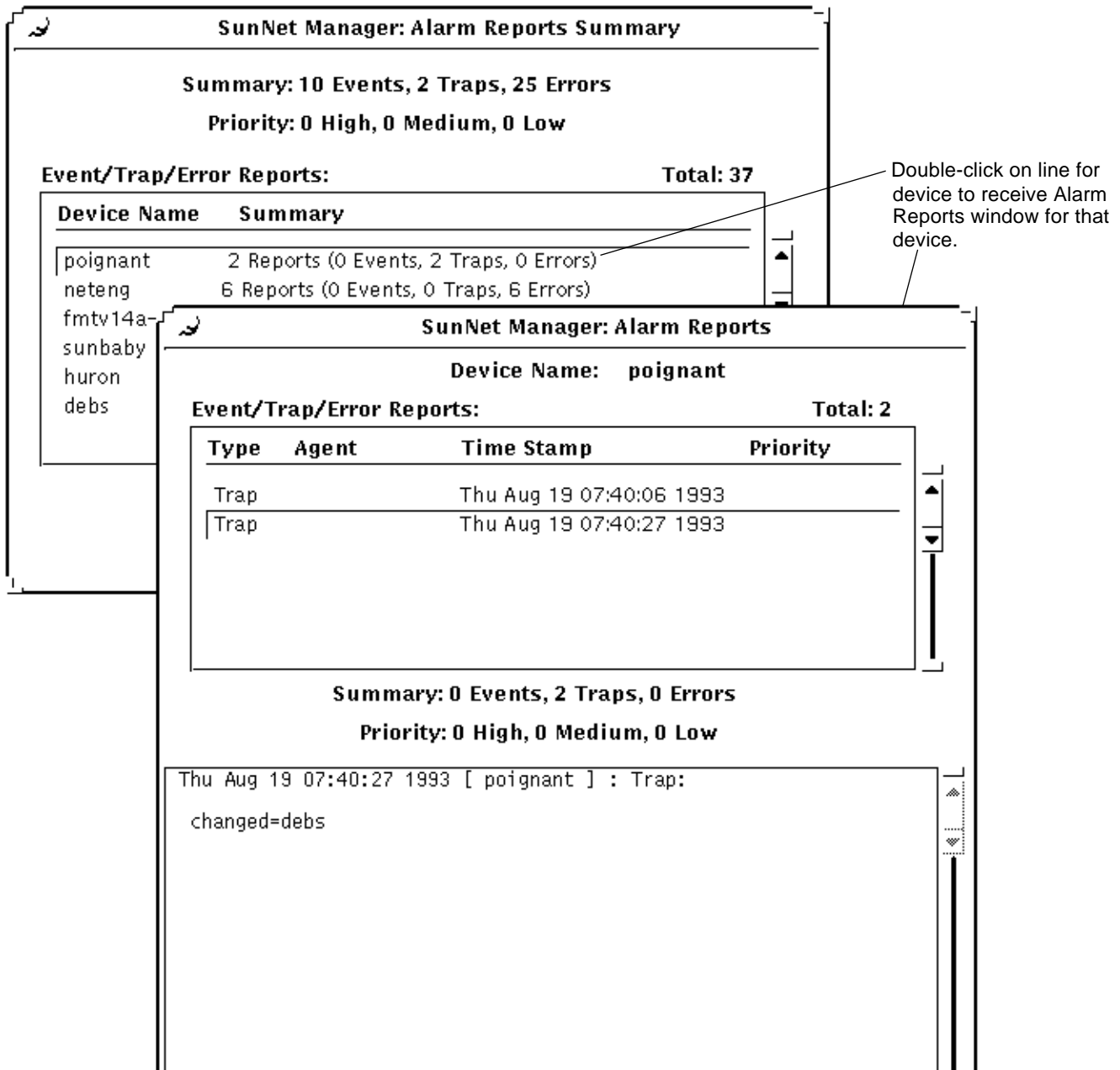


Figure 6-4 Device-specific Alarm Reports Window

6.0.4 Device-specific Alarm Reports Window

A device-specific Alarm Reports window allows you to:

- Sort alarm reports
- Filter reports
- Find reports associated with a specific agent
- Save reports to a file
- Print reports

These functions are available through the View, Save, and Print buttons at the bottom of the Alarm Reports window. The functions are described in detail in “Part 2: Reference.”

6.0.4.1 Filtering Alarm Reports

1. Press **MENU** in the View button, to receive the following menu;

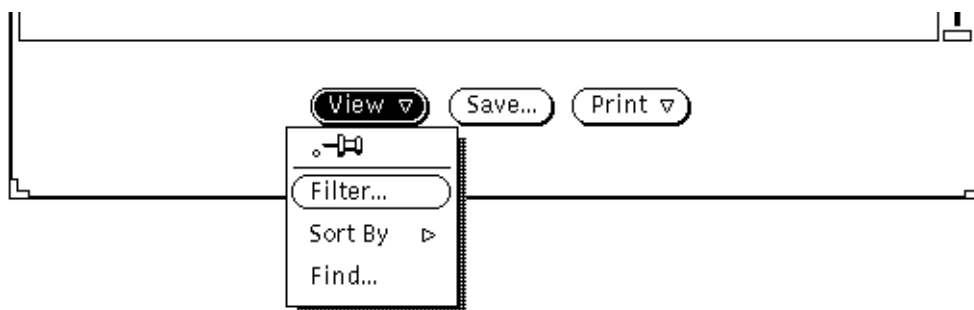


Figure 6-5 Device-specific Alarm Reports View Menu

2. Release **MENU** over Filter to receive the following window:

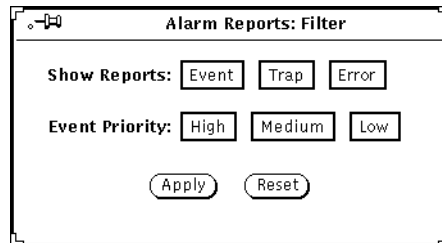


Figure 6-6 Device-specific Alarm Reports Filter Window

Your selections in the Filter window determine which types of reports are displayed in the device-specific Alarm Reports window. The default is that all alarm reports are displayed. Priority choices apply only to Event Reports.

- Click on Apply, your choices take effect immediately.
- Click on Reset to restore the choices to the way they were when you opened the Filter window.

Note – If you plan to both sort (described below) and filter your device-specific alarm reports, you must sort before filtering. If you want to sort again, after filtering, you must click SELECT on Reset, then Apply, in the Filter window, sort again, as described below, then reinvoke Filter.

6.0.4.2 *Sorting Alarm Reports*

1. In a device-specific Alarm Reports window, press MENU on View►Sort By. You receive the following menu:

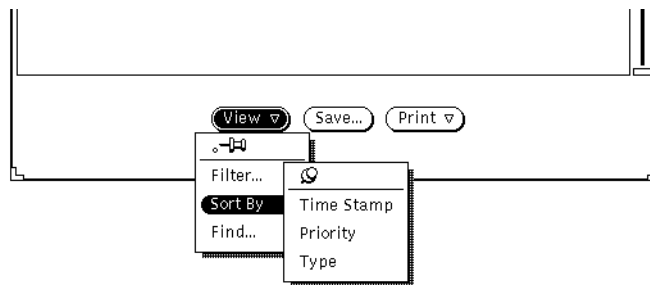


Figure 6-7 Device-specific Alarm Reports Sort Menu

2. Release MENU over the sorting method you want. Your selection takes effect immediately. The choices are mutually exclusive.

The sorting choices are:

Time Stamp: The default. Reports are sorted in chronological order with the most recent appearing last on the list.

Priority: high priority reports are listed first. Use this option if you are filtering out trap and error reports. If you are not filtering out reports and you specify sorting by priority, event reports are listed first, followed by error reports, then trap reports.

Type: Reports are sorted by type, in the order: event, error, trap. Note that alarm reports received after you specify sorting by type, are appended to the appropriate list of reports, depending on the type of the new reports.

Note – When sorting alarm reports, only the reports currently displayed in the Alarm Reports window are sorted. All new reports received during the sorting operation are appended to the end of the sorted list.

6.0.4.3 Finding Agent-specific Alarm Reports

You can search on an agent name to find instance(s) of event reports originating with that agent. To do this, in the device-specific Alarm Reports window, press MENU on View►Find and release MENU to receive the window shown in the Figure below.

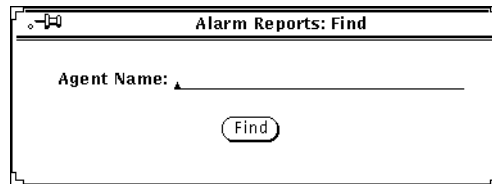


Figure 6-8 Device-specific Alarm Reports Find Window

In the Agent Name field, enter the agent *and* group name—not just the agent name. For example, specify `hostperf.data`, not just `hostperf`. When you click SELECT on Find, the window displays the contents of the first (oldest) report originating with the agent you specified. If there is no report originating with the agent you specify, you receive the message, “Agent name not found.”

6.0.4.4 Saving Device-specific Alarm Reports

SunNet Manager allows you to save device-specific alarm reports to a file. To do this, in the device-specific Alarm Reports window, click SELECT on Save. You receive a window such as the one in Figure 6-9.

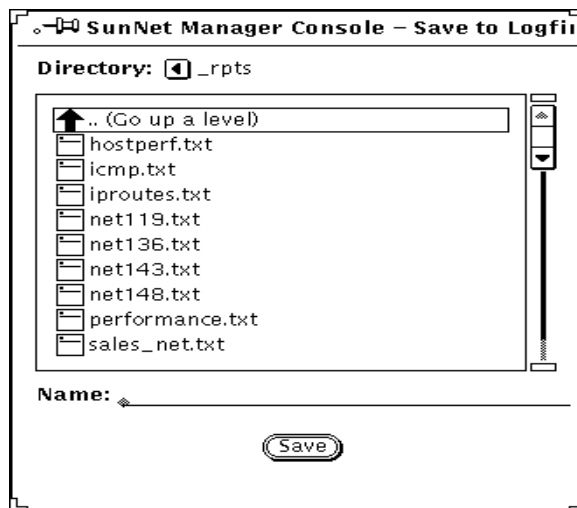


Figure 6-9 Save to Logfile Window

If the directory first displayed is not the directory you want, change directories by clicking SELECT on folder icons or the . (level above) symbol. Alternatively, you can enter a pathname in the Name field and click SELECT on Save. If the path you enter ends in a valid directory, you are switched to that directory.

When you reach the directory you want, enter the name of the file to which the report will be written.

6.0.4.5 Printing Device-specific Alarm Reports

SunNet Manager allows you to print device-specific alarm reports. Click SELECT on the Print button on the bottom of the device-specific Alarm Reports window, or press MENU on Print►Report and release MENU. SunNet Manager sends the currently displayed alarm report to your default printer.

6.1 Viewing Error Messages and Error Reports

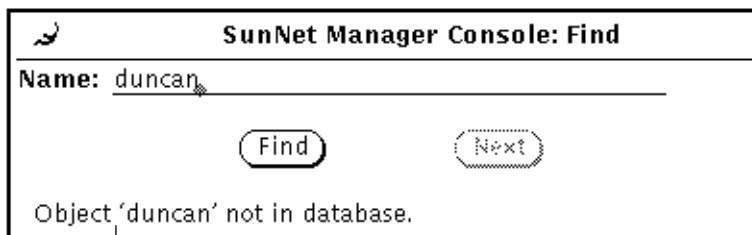
The Console displays error messages in several locations:

- Footer of the Console window.
These are generally brief messages about the status of an operation and are *temporary*. They are overwritten by subsequent status messages. Because of the brevity of these messages, you are usually directed to the Error Reports window for more information.



Figure 6-10 Console Window Footer Display

- Footer of the Console pop-up windows or a tool (Results Browser, Grapher, Set Tool) window.
These messages are described in the *Site/SunNet/Domain Manager Troubleshooting Guide*.



Pop-up window footer message.

Figure 6-11 Console Pop-up or Tool Window Footer Display

- Error Reports window.
Errors detected by the Console and errors returned by agents are displayed in this window. These messages are described in *Site/SunNet/Domain Manager Troubleshooting Guide* and in the man page for the agent.

To view the Error Reports window:

1. Move the mouse pointer over the View button and press MENU to open the View menu.
2. Drag the mouse pointer down to Error Reports and release MENU. You receive a sample error reports window similar to the one below:

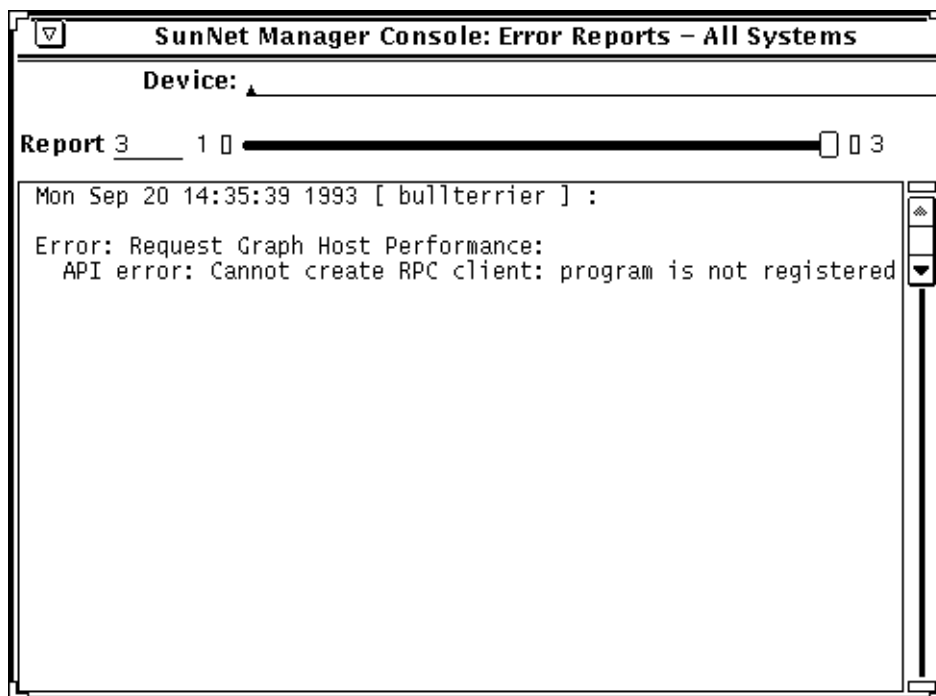


Figure 6-12 Sample Error Reports Window

3. To examine error reports for a particular system, type in the device name after the Device prompt and press Return.
To see entries for all elements again, press Ctrl-U to clear the Device line and press Return.
4. To save the error reports to a file, press MENU in the output portion of the Error Reports window and release MENU over File►Save or File►Store as New File. In the window you then receive, select or enter a directory and enter a file name. Click SELECT on the Save button.
5. To browse through entries in the log:

- Press SELECT on the slider and drag the slider to the left or right.
- Click SELECT to the left or right of the slider.
- Click SELECT on either end of the slider bar.
- Enter the number of a report on the Report line and press Return.

6.1.1 Useful Notes about Types of Errors

Two types of agent errors are returned to the Console. Agent-specific errors are errors defined in the agent schema file—these error messages are described in the `man` pages for each agent. There are also “generic” errors, such as “unknown host” or “no threshold value”—these errors are described in the *Site/SunNet/Domain Manager Troubleshooting Guide*.

Agent errors can also be classified as “fatal” or “warning.” All generic errors are fatal errors. Agent-specific errors can be either fatal or warning. Fatal errors cause the agent to stop servicing the request. Warnings are for your information only; the agent will continue to return reports.

The Errors category of the Console Properties window allows you to specify signal options and any operations to be started when the Console receives an error from an agent. The default Errors category settings cause the glyph for the target element to be dimmed only if a fatal error is reported for the element. The glyph effect is not propagated. You can change the Errors category settings—for more information, refer to “Part 2: Reference.”

Fatal errors reported by agents, whether agent-specific or generic, are considered in the Console as high-priority errors. The Console treats warnings as low-priority errors. Thus, if you specify Color by Priority in the Glyph Effect setting for the Errors category in the Console Properties window, fatal errors will change an element glyph to red, and warnings will change an element glyph to yellow, or to colors you have customized.

By default, a maximum of 100 error reports are displayed in the Error Reports window. You can change the maximum number in the Console Properties window. See the description of the Maximum Error Reports setting in “Part 2: Reference.”

6.2 Viewing Traps

A trap is an unsolicited report sent from an agent that usually signifies some unexpected error condition. Trap reports are displayed in the Event/Trap Reports window.

When a trap is generated, a trap report stating the cause of the trap is returned to the Console. The report is stored in the `event.log` file and may be viewed using the following steps.

1. **In the Console window, press MENU in View►Event/Trap Reports and release MENU.**

You receive an Event/Traps Reports window similar to the one below.

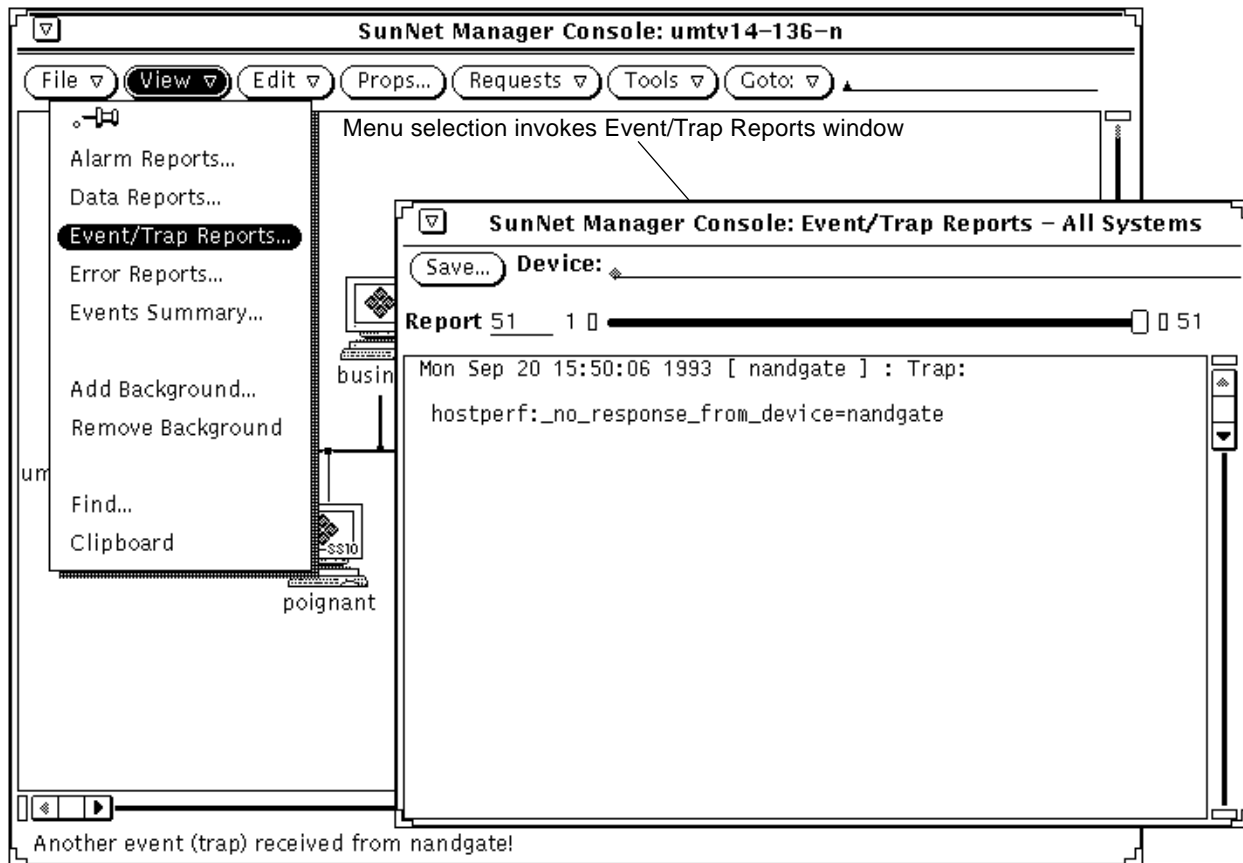


Figure 6-13 View—Event/Trap Reports Window

2. To examine trap reports for a particular system, type in the device name after the Device prompt and press Return.

To see entries for all elements again, press Ctrl-U to clear the Device line and press Return.

An alternative way to obtain an element-specific trap reports is to click SELECT on the glyph in the Console window, then invoke View►Event/Trap Reports.

3. **To save the Event/Trap Reports to a file, click SELECT on the Save button.**
A pop-up Save window appears, where you can enter the path and file name where the event/trap reports are to be stored.
4. **To browse through entries in the log:**
 - Press SELECT on the slider and drag the slider to the left or right.
 - Click SELECT to the left or right of the slider.
 - Click SELECT on either end of the slider bar.
 - Enter the number of a report on the Report line and press Return.

6.2.1 Useful Notes about Report Settings and Display

The Trap Information settings in the Events and Traps category of the Console Properties window allow you to specify signal options and any operations to be started when the Console receives a trap. The default Events and Traps category settings cause the glyph for the target element to blink only if a trap is reported for the element. By default, the glyph effect is propagated. If the Console window is closed to an icon, you can specify that it be opened automatically if an event or trap is received. To view or change event- and trap-related settings, click SELECT on Props in the Console's control area, then select Events and Traps in the Category pulldown menu. For more information on changing Trap Information settings, refer on the Console's Properties window in "Part 2: Reference."

Trap reports are generated when elements are added, changed, or deleted in the runtime database. By default, these traps are ignored by the Console. If you want to have these traps displayed, toggle the check mark off in the Ignore Database Traps setting for the Events and Traps category. If a glyph effect is specified for traps, the glyph representing the system on which the runtime database is located is affected.

Note – When you create an element with the Ignore-Database-Traps option turned off in the Console's Props►Events and Traps window, the trap report for that element says that the element was "created." An analogous operation using the database API returns the word "added" rather than "created."

6.2.2 *Priority Settings for Traps*

Traps can be high, medium, or low priority. Starting with version 2.3 of SunNet Manager, you can determine priority level for any SNMP trap (low, medium or high). Enter your choice into the trap configuration file, `snmp.traps` after the trap description field. See Chapter 8, “Managing SNMP Devices” for information on how to set SNMP trap priorities. Traps generated by changes to the runtime database are considered low-priority.

If you specify Color by Priority in the Glyph Effect setting for the Events and Traps category in the Console Properties window, SNMP traps will change an element glyph to the default color of red, or to the color you have customized. Database traps will change an element glyph to the default color of yellow or to the color you have customized. For more information about customizing Color by Priority, see “Part 2: Reference.”

By default, a maximum of 1000 trap reports are displayed in the Event/Trap Reports window. You can change this maximum number in the Console Properties window. See the description of the Maximum Trap Reports setting in “Part 2: Reference.”

Managing Printers



This chapter discusses the following topics:

- Checking the queue of a remote printer
- Checking the status of a remote printer

Through the `lpstat` agent, SunNet Manager allows you to check the queue and check the status of remote printers. A printer can be connected to a machine running SunOS 4.x or Solaris 2.x.

Perform the following steps to manage a remote printer:

1. Install agents on the machine to which the printer is connected (the printer server).

If the machine is running Solaris 2.x, use `pkgadd`; if it is running SunOS 4.x, use `getagents`. See your installation guide for instructions on installing agents.

2. In the Console window, create an element of category component and type `laserwriter`, `newsprinter`, or `printer`, as appropriate.

See the chapter on “Creating and Modifying the Management Database” for instructions on creating an element. Figure 7-1 shows an example Create Object window.

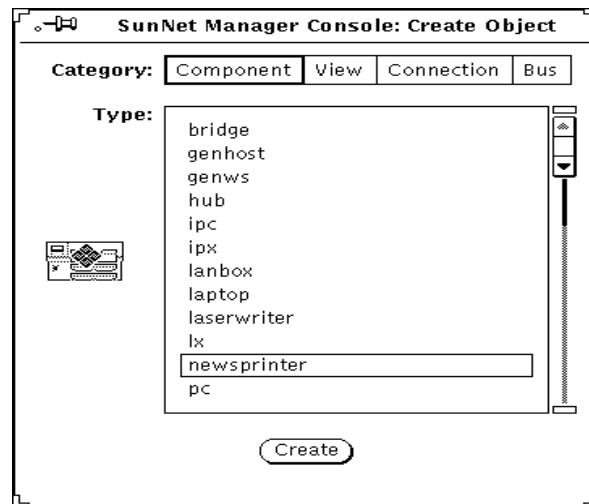


Figure 7-1 Creating a Printer

3. After clicking SELECT in the window above, you receive an Element Properties window. In this window, do the following:

- a. Enter the name of the printer (print queue name)—*not* the name of the printer server.
- b. Check off the `lpstat` agent.
- c. Enter the name of the connected machine as the proxy.
- d. Click SELECT on Apply.

Do not check off any agents other than `lpstat`. See Figure 7-2 for an example Element Properties window.

SunNet Manager Console: New component.newsprinter

Name:

Contact:

Location:

Description:

ippath IP path information

iproutes IP route table and statistics

layers protocol layer statistics

layers2 protocol layer statistics (for SunOS 5

lpstat master line printer status and queue info

ping IP connectivity info

rpcnfs RPC and NFS stats

sample kernel mbufs agent

snmp vanilla snmp agent

snmp-mihll snmp-mihll proxy

Red: 0

Green: 0

Blue: 0

Create

Figure 7-2 Properties for New Printer

If you want to receive data or event reports from the printer server, you must create a separate element for that machine. When specifying agents for the printer server, do not check off the `lpstat` agent.

You can obtain Quick Dumps and send data and event requests to the printer, just as you would send such requests to a workstation, server, or other type of element. SunNet Manager is shipped with a predefined event request so that you can be informed if the printer becomes unavailable. Figure 7-3 shows how to submit this request.

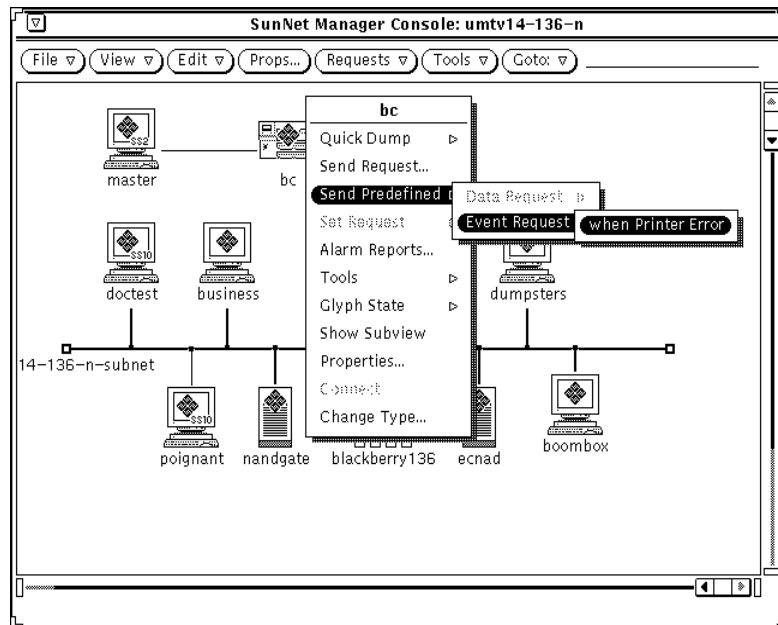


Figure 7-3 Sending Predefined Request to Printer

Through the Quick Dump mechanism (in the printer's glyph menu press MENU on Quick Dump►lpstat►queue or ►status), you can view the current job queue for a printer or obtain the current status. Status is reported as one of the following codes:

- 0 idle
- 1 waiting
- 2 printing
- 3 error

Through the lpstat agent, you can obtain a variety of information about print queue and printer status. There are some differences between information returned from printers connected to SunOS 4.x printer servers and that returned from Solaris 2.x print servers. See the `na.lpstat(8)` man page for details.

Managing SNMP Devices



This chapter discusses the following topics:

- Adding SNMP devices
- Creating an SNMP element in the database
- Setting up SNM to receive traps from a device
- Using the Set Tool to retrieve SNMP attribute values
- Using the Set Tool to change SNMP attribute values

SunNet Manager provides a proxy agent that supports the Simple Network Management Protocol (SNMP). The *SNMP proxy agent* allows you to get data and event information from, and set attribute values for, devices that are manageable through SNMP. The proxy agent can handle requests from the Console for multiple SNMP devices on a network.

The *SNMP trap daemon* receives traps from SNMP devices, filters the traps, and translates the traps into SunNet Manager traps. The trap daemon then forwards the traps to one or more management stations.

Management of SNMP devices is similar to management of non-SNMP devices, with the following exceptions:

- Multiple schemas can be associated with an SNMP device (however, only one schema can be specified in a request).

- While SunNet Manager uses schema files to describe the attributes of a managed object, the SNMP protocol uses a mib to describe the attributes of a managed object.

This chapter describes how to integrate SNMP devices into your SunNet Manager environment in order to use SNMP to manage those devices. You should have some familiarity with SNMP concepts to proceed with these tasks. Refer to “Part 2: Reference” for a detailed description of how the SNMP proxy agent works.

8.1 Adding SNMP Devices

The following is an overview of how to add non-Sun SNMP devices to your management database.

If you are using SNMP to manage Sun workstations that contain the SNM-supplied SNMP agent (`snmpd`), you can skip this task and proceed to the section on “Creating an SNMP Element in the Database.”

1. Install the device on the network.

2. Obtain the schema file for the device.

3. Configure the SNMP proxy agent.

The schema file should reside in a directory specified by `na.snmp.schemas` in the `snm.conf` file, for example:

```
# @(#)snm.conf2.36 6/30/96 - SunNet Manager configuration file
# Copyright (c) 1990,1993,1996 by Sun Microsystems Inc.

# Site-specific configuration information

### Keywords for the SNMP proxy agent:
# Directory list for SNMP schema files. Separate each directory
# with a colon.
na.snmp.schemas /opt/SUNWconn/snm/agents
```

4. On the Console system, load the schema file.

Skip this step if you are using the SNMP schema files provided with SNM (`snmp.schema` for MIB I devices, or `snmp-mibII.schema` for MIB II devices). The schema file should reside in a directory specified in the SNM Console Properties window.

5. Create the device in the Console.

See “Creating an SNMP Element in the Database” for information about this step.

The following is a detailed explanation of each step:

1. Install the device on the network.

Refer to the device documentation for installation information. If the device sends SNMP traps that you want forwarded to the SNM Console, you will also want to set up the SNMP trap daemon and Console for trap handling. See “Setting Up SNM to Receive Traps from a Device” for more information.

2. Obtain the schema file for the device.

SunNet Manager uses schema files to manage objects. You can obtain a schema file in one of several ways:

- The vendor of the SNMP device may provide a SunNet Manager schema file. If the SNMP device sends enterprise-specific traps, the vendor may also supply a file that describes these traps—see “Setting Up SNM to Receive Traps from a Device” for more information about setting up SNM to handle enterprise-specific traps.
- The `snm-server@sun.com` electronic mail server maintains a collection of schema files for various SNMP devices. For information on accessing the mail server, see “SunNet Manager Electronic Mail Distribution Service” in the Preface of this book.
- If the SNMP device adheres to MIB I or MIB II specifications, you can use one of the schemas supplied with SNM—either `snmp.schema` (for MIB I) or `snmp-mibII.schema` (for MIB II). If you installed the SNMP proxy agent with `getagents`, the Sun-supplied SNMP schemas are automatically copied onto the host system with other schemas.
- Starting with version 2.3, SunNet Manager schemas for several RFC MIBs are bundled.
- If you do not have a schema file for the device that you want to manage, you can create a schema file from the MIB for the device with the `mib2schema` utility. Refer to “Part2: Reference” for more information.

Change the name of the schema, if necessary. The schema name is *not* the schema file name. The schema name is specified by the keyword `proxy` in the schema file. If `mib2schema` was used to generate the schema file, this name is derived from the name of the MIB (not the MIB file name). This is

the name that will appear in the list of agent schemas for each element. Therefore, you may want to edit this name to ensure that it is meaningful to you.

3. Configure the SNMP proxy agent.

- a. A single proxy agent can handle management requests for many devices. You can allow the SNMP proxy agent on the manager station to handle all SNMP requests. However, you may want to distribute the proxy agent to minimize network traffic—see “Management Applications and Agents” for more information.
- b. On the proxy system, make sure that the schema file is available to the SNMP proxy agent. The schema file should reside in a directory specified by the `na.snmp.schemas` keyword in the `snm.conf` file on the proxy system. By default, the `snm.conf` file is located in the following directory:
 - `/etc` for the Solaris 1.1.1 version of the current SunNet Manager product
 - `/etc/opt/SUNWconn/snm` for the Solaris 2.x version of the current SunNet Manager product

Note – If you modify the `snm.conf` file while the SNMP proxy agent is running, you need to kill the proxy agent (`na.snmp`) for the changes to take effect. Refer to the *Solstice Site/SunNet/Domain Manager Troubleshooting Guide* for information on how to kill the agent.

4. On the Console system, load the schema file.

Skip this step if you are using the SNMP schema files provided with SNM (`snmp.schema` for MIB I devices, or `snmp-mibII.schema` for MIB II devices).

To load the schema file into the Console:

- a. Start the Console.
- b. Use the Load►Management Database option in the File menu to load the schema file.
- c. Make sure that the schema file is located in a directory that is specified by the Schema Directories setting in the Console Properties window.

8.2 Creating an SNMP Element in the Database

The IP Discover Tool automatically creates SNMP elements. You can specify an option in IP Discover that allows you to discover only SNMP elements or SNMP elements in combination with other types of elements. See “Part 2: Reference” for a description of the IP Discover Tool.

To create SNMP elements one-at-a-time, use the Console’s graphical editor. Using this method, creating an element that will be managed with the SNMP proxy agent is essentially the same as creating any element in your database. To define an SNMP element instance in a Console view, you specify SNMP-specific information about that element in the Properties window.

1. **Move the mouse pointer to the Edit button and press MENU to open the Edit menu. Release MENU over the Create option. You receive the Create Object window:**

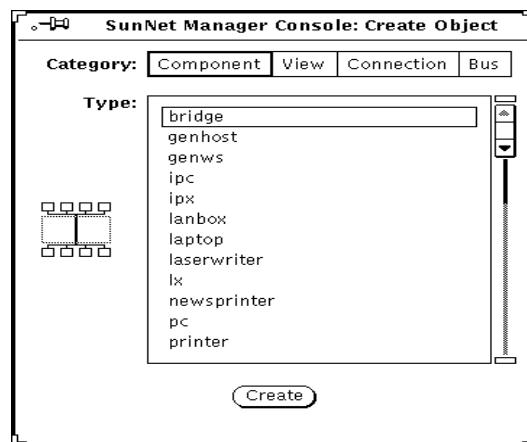


Figure 8-1 Edit—Create Window

2. **Click SELECT on the category of element and type of element you want. Then click SELECT on Create.**
3. **Fill in the Properties window for the element.**
The top portion of the Properties window is the element data. You must fill in the Name field; other fields are optional. Specify the SNMP Read Community and SNMP Write Community fields for the element. Although

filling in the SNMP fields (including SNMP Vendor Proxy and SNMP Timeout) is optional, you should understand how the proxy agent uses this information—see the discussion below.

The middle portion of the Properties window is the list of agent schemas that the Console knows about. Check the schemas that apply to the device. There are three SNMP schemas supplied with SunNet Manager: `snmp` describes the MIB I, `snmp-mibII` describes MIB II, and `sun-snmp` is the schema file for the SNMP agent (`snmpd`) for Sun workstations. If you loaded any additional schemas into the Console, the names of these schemas appear in the schemas list. Note that the schema name is *not* the schema file name. The schema name is specified by the keyword `proxy` in the schema file. See “Adding SNMP Devices” for more information about setting the schema name.

4. Click SELECT on the Apply button in the properties window.

A glyph for the element you just defined appears with the name you specified.

Figure 8-2 shows a properties window for a new component that has the SNMP schemas checked.

SunNet Manager Console: New component.ss2

Location: _____
Description: _____
SNMP RdCommunity: _____
SNMP WrCommunity: _____
SNMP Vendor Proxy: _____
SNMP Timeout: 0 [▲▼]

Ipstat line printer status and queue info
 ping IP connectivity info
 rpcnfs RPC and NFS stats
 sample kernel mbufs agent
 snmp vanilla snmp agent
 snmp-mibII mibII_test snmp-mibII proxy
 sun-snmp Sun SNMP agent
 sync synchronous interface stats
 traffic IP traffic analyzer

Red: 0 [Slider]
Green: 255 [Slider]
Blue: 0 [Slider]

Apply Reset Alias... Create

Figure 8-2 Properties Window for a New Component

Field Values

The values in the fields SNMP Read Community, SNMP Write Community, SNMP Vendor Proxy, and SNMP Timeout are sent with the request to the SNMP proxy agent. If you do not specify values in the SNMP Read Community, SNMP Write Community, and SNMP Timeout fields, the SNMP proxy agent uses the following values:

- Read community is “public.”
- Write community is “public.”

- Timeout will be the value (in seconds) specified by the keyword `na.snmp.request_timeout` in the `snm.conf` file on the system where the SNMP proxy agent resides. The keyword's supplied value is 5 (seconds).

Vendor Proxy Field

SNMP Vendor Proxy is an optional field that specifies the name of a proxy system with which the SNMP proxy agent will communicate. Normally, you do not need to specify a value for this field. If this field is used, the SNMP request is passed through the element to a secondary proxy. This field should *only* be specified when a vendor has supplied an SNMP proxy agent to manage a particular device or set of devices that do not support IP connectivity. The vendor's SNMP proxy agent communicates with the SunNet Manager SNMP proxy agent through SNMP, but communicates with the element using either SNMP or a different protocol.

Agent Schema Files

You can select multiple SNMP agent schema files for an element. However, only one schema is associated with each request. Merely checking an SNMP agent schema on the Properties window does not make the element manageable through the SNMP proxy agent. SNMP agent software must be installed and running on the SNMP device. The MIB for the device must contain the same data definitions as the schema file used by the Console and the SNMP proxy agent.

The blank lines next to the SNMP agent schema names allow you to specify the name of the system on which the SNMP proxy agent resides. This system name is for a default proxy system. You can specify a different proxy system name for each request. If you do not specify a proxy system in the Properties window or in the report request, the system on which the Console is running is assumed to be where the proxy agent resides.

8.3 Setting Up SNM to Receive Traps from a Device

Many SNMP devices send out unsolicited or unexpected reports called traps. For example, a trap may be sent when a device is restarted. When the SNMP trap daemon receives traps, it will forward all or only selected traps to one or more SNM Consoles.

- 1. Determine the location of the SNMP trap daemon.**

See the discussion later in this section.

- 2. If you want to define the priority of traps or discard traps based on enterprise-specific traps or host-specific traps, create file entries for these traps.**

For example:

```
#
# Example traps
enterprise 1.3.6.1.4.1.42
    1      CPU_Failure      high
    2      Power_Supply_Failure      medium
    3      Network_Connection_Failure      low
    4      Over_Heating      discard
    5      RealTimeClock_Failure      discard
```

Setting Enterprise-Specific Trap Priorities

Starting with version 2.3 of SunNet Manager, you can assign low, medium, or high priority to an enterprise-specific SNMP trap. Enter the priority into the trap configuration file (snmp.traps) after the trap description field. For example:

```
#
# Sample traps
host shanghai
    1      CPU_Failure      high
    2      Power_Supply_Failure      medium
    3      Network_Connection_Failure      low
    4      Over_Heating      discard
    5      RealTimeClock_Failure      discard
```

If you choose to discard all traps from an enterprise or assign a priority to all traps for an enterprise, you can specify this on the keyword values next to the Object Identifier following the enterprise keyword. For example:

```
#
# Sample traps
      1      snmp.traps enterprise 1.3.6.1.5.1.75      discard
```

The snm.conf file includes the following priority keyword:

```
na.snmp.trap.default-priority
```

The default value is low; however, any of the three values can be used. If you change the value, stop and restart the trap daemon, na.snmp-trap. An enterprise specific trap priority overrides the priority specified by na.snmp.default-priority in the snm.conf file.

Setting Host Specific Trap Filters

Starting with version 2.3, you can specify filters based on certain hosts. What you specify overrides any enterprise-specific filter. The keyword, <host>, can be used in the snm.conf file followed by the ip_address or host name and a priority keyword. Each line beginning with the keyword <host> can be followed by subsequent lines describing the action, description, or priority for each trap. For example:

```
<host> 139.146.75.165 discard
<host> 149.136.75.200 medium
      6 TEMP_HIGH high
      10 PORT_TEST_FAIL discard
```

Precedence Values

The snm.conf file specifies general priority for all traps. The trap daemon will compare oid entries with specific settings specified in snmp.traps when it receives a trap. If a particular trap has an oid entry in snmp.traps, the na.snmp-trap daemon will take the priority of the enterprise trap. The na.snmp-trap daemon searches for the hostname of the target trap. If the name is present, the daemon will use it as this trap’s priority.

3. Specify where the trap daemon should forward the traps.

Forwarding Traps

Starting with version 2.3 of SunNet Manager, you can forward all raw trap PDU packets to other workstations or to a different port on the same machine using the keyword `na.snmp-trap.forward`. You can specify a maximum of two hosts and the appropriate UDP ports to which the trap PDUs are to be forwarded.

Add the keyword `na.snmp-trap.forward.snmp-traps: <hostname>, <port>: <hostname>, <port>` in the `snm.conf` file. You can specify more than one host name—separate each host name with a colon (:).

Use the keyword `na.snmp-trap.rendez` to forward a SunNet Manager format SNMP trap to one or more consoles. Add the keyword to the `snm.conf` file.

If you modify the `snm.conf` file while the SNMP trap daemon is running, you must kill the trap daemon (`na.snmp-trap`) for the changes to take effect. For example:

```
### Keywords for the SNMP trap proxy on wordstwo:
# default file name of per-enterprise traps
na.snmp-trap.default-trapfile /var/adm/snm/snmp.traps
#
na.snmp-trap.rendez rubicon
```

(Refer to the *Solstice Site/SunNet/Domain Manager Troubleshooting Guide* for information on how to kill the daemon.)

4. On the SNMP device, specify that traps be sent to the host where the trap daemon resides.

See the discussion below. Typically, you would specify the IP address of the host system. However, this is very device-specific. Refer to the device documentation to find out how to do this.

Warning – Use SNMP trap forwarding to forward SNMP traps with caution. If you forward traps within the SunNet Manager environment, and infinite loop (machine A->machine B->machine A) may occur, exhausting machine resources in the process and increasing the burden on network traffic. If an infinite loop should occur, see the *Solstice Site/SunNet/Domain Manager Troubleshooting Guide* for resolution.

Trap Messages

A single trap daemon can accept trap messages from many devices. The trap daemon does not need to reside on the SNMP proxy agent system or the Console system. Like proxy agents, the SNMP trap daemon can be distributed in a network to reduce network traffic and distribute the processing load between systems. See Chapter 1, “Overview and Concepts” for more information. For example, you might want to have traps from all of the SNMP devices on a subnet sent to one host system. The trap daemon on this system can then forward *some* or *all* of these traps to selected Console system(s).

Types of Traps

The SNMP protocol defines six generic types of traps. In addition to returning generic traps, an SNMP device may return enterprise-specific traps, which are defined by the device vendor. (See the device documentation for information on enterprise-specific traps.) Enterprise-specific traps are usually defined in the device MIB. If the MIB for the device contains enterprise-specific traps and you used the `mib2schema` utility to generate a schema file, a trap file, such as `snmp.traps`, is automatically generated. This file may be copied into place or its contents can be appended to the existing trap file. (See the `mib2schema(1)` man page for more information.) Otherwise, create an `snmp.trap` file as described in “Part 2: Reference.”

Discarding Traps

You may determine that certain generic or enterprise-specific traps do *not* need to be forwarded to the Console. For example, you may decide that a trap does not need to be sent to the Console every time a device is warm-started. You can add the optional keyword “discard” to trap definition entries in the trap file—see the trap file syntax in “Part 2: Reference.” The default trap file contains trap definitions according to enterprise ID. If multiple devices on the network use the same enterprise ID and all the trap file entries apply to each device, add the trap entries to the default trap file. The default trap file is specified by the keyword `na.snmp-trap.default-trapfile` in the `snm.conf` file on the system on which the trap daemon resides. Normally, the default trap file name is:

- `/var/adm/snm/snmp.traps` for the Solaris 1.x version of the current product.
- `/var/opt/SUNWconn/snm/snmp.traps` for the Solaris 2.x version of the current product.

Trap Priorities and Filters

See “Setting Trap Priorities” and “Setting Trap Filters” earlier in this chapter for information about these features.

If multiple devices on the network use the same enterprise ID, but the trap file entries apply to only specific device names, define an entry for each device in the SNMP host file. Refer to “Part 2: Reference.”

8.4 Using the Set Tool to Retrieve SNMP Attribute Values

You can use the Console to obtain Quick Dumps and data, event, and trap reports on SNMP devices, just as with any other type of device. See the following chapters for steps to get this information: Chapter 4, “Requesting Data,” Chapter 5, “Specifying Event Requests” and Chapter 6, “Viewing Reports.”

For SNMP devices you can also use the Set Tool to retrieve SNMP attribute values. Retrieving attribute values with the Set Tool allows you to examine an attribute’s current value before you change it—see Section 8.5, “Using the Set Tool to Change SNMP Attribute Values,” on page 8-18 for information on changing attribute values.

- 1. Move the mouse pointer over the glyph that represents the target SNMP device and press MENU to open the Glyph menu.**
- 2. Drag the mouse pointer down to Set Request. Drag right over the desired agent schema name and continue to drag right over the desired group or table name. For example:**

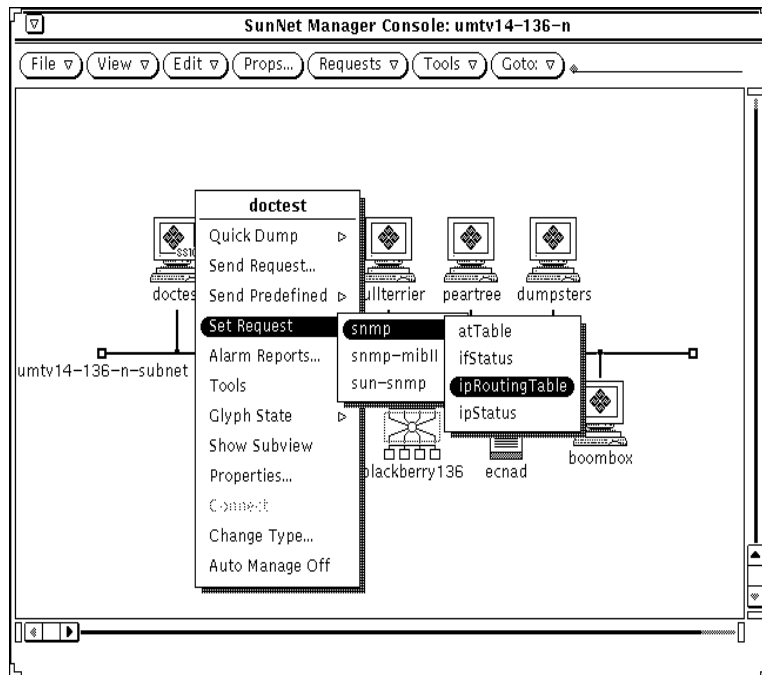


Figure 8-3 Glyph—Set Request Menus

In your response to your menu selection, you receive a window such as the one shown in Figure 8-4.

SunNet Manager – Set : augusta

Get Set Unset

Agent ▾ snmp

Group ▾ ipRoutingTable

Key ▾ _____

Options : _____

Attribute Name	Current Value	New Value	
ipRouteDest		_____	Details...
ipRouteIfIndex		_____	Details...
ipRouteMetric1		_____	Details...
ipRouteMetric2		_____	Details...
ipRouteMetric3		_____	Details...
ipRouteMetric4		_____	Details...
ipRouteNextHop		_____	Details...
ipRouteType		_____	Details...

Set Information: File ▾ Delete ▾

Figure 8-4 Set Tool Window

3. In the Set Tool window, specify the key or options for the request.

If the group is a table and you know the key of the row you want, type in the key value in the Key field. If you do not specify a key value, when you SELECT the Get button (in the next step) the first row of the table will be

displayed by default and the Key menu generated. You can then choose a value from the Key menu by pressing MENU on the abbreviated menu button.

You can specify an SNMP read-community name in the Options field if it is different from the read-community you have previously specified in the Properties window for the element.

4. Click SELECT on the Get button. The attribute values are displayed in the attribute list.

You can see detail on an attribute by clicking SELECT on the Details button. Figure 8-5 is an example of information available.

Name: is the name of the attribute

Type: is the type of the attribute. If the attribute is an enumeration, the valid values will be displayed.

Access: indicates what access categories are assigned to the attribute

Description: gives the meaning of the attribute

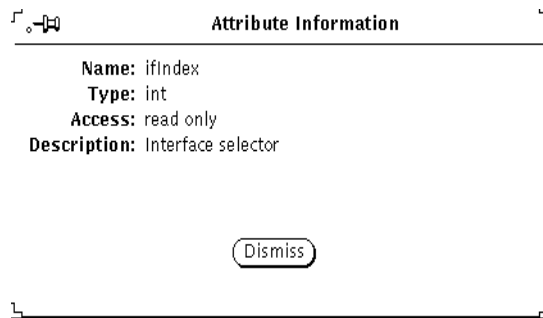


Figure 8-5 Attribute Information Screen

8.4.0.1 Retrieving Other Attribute Values

In the control panel at the top of the Set Tool window, you can change the attribute group or agent schema name (or both) to retrieve attribute values for other attribute groups or for other agent schemas used by the target device.

- To change the agent schema:

- a. Move the mouse pointer over the Agent abbreviated menu button.
- b. Press MENU to open the list of agent schemas available for the target element.
- c. Drag the mouse pointer to the desired agent schema and release.

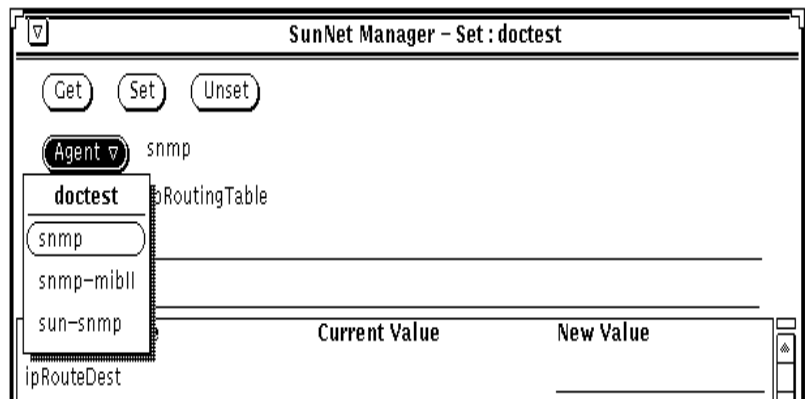


Figure 8-6 Set Tool—Agent Menu

- To change the attribute group:
 - a. Move the mouse pointer over the Group abbreviated menu button.
 - b. Press MENU to open the list of attribute groups specified in the agent schema.
 - c. Drag the mouse pointer to the desired group and release.
 - d. If the group is a table and you know the key of the row you want, type in the key value in the Key field.

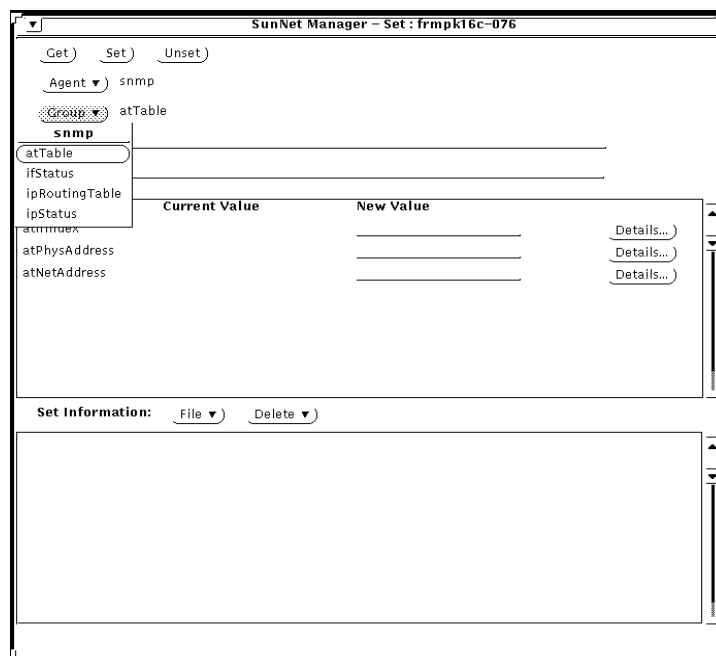


Figure 8-7 Set Tool—Group Menu

8.5 Using the Set Tool to Change SNMP Attribute Values

A set request is used to request an agent to change the value of a particular attribute. With one Set Tool operation, you can send set requests to different agents for multiple attributes with one Set Tool operation.

1. Move the mouse pointer over the glyph that represents the target SNMP device and press MENU to open the Glyph menu.
2. Drag the mouse pointer down to Set Request. Drag right over the desired agent schema name and continue to drag right over the desired group or table name. You receive a Set Tool such as the one shown in Figure 8-4 on page 8-15.

3. In the Set Tool window, specify the key or options for the request.

If the group is a table, type in the key value in the Key field. If you do not know the key, retrieve the available key values by clicking SELECT on the Get button.

4. Enter the new value on the line provided in the New Value column and press Return. The new value is shown in the lower portion of the window.

5. To change another attribute in a different attribute group or agent schema, use the Group and Agent abbreviated menu buttons.

All the attributes in the selected group appear in the attribute list. If you want to change the value of an attribute of a row in a table, you must supply the key for that row. You can also click the Get button to display the current values of the attributes.

You can specify an SNMP write-community name in the Options field if it is different from the write-community you have previously specified in the Properties window for the element.

6. Repeat the previous three steps until you have entered all the attribute changes you want to make for the target element.

Note that the new attribute values are collected in the Set Information list. This allows you to view your changes and edit them, if necessary, before the actual set request is made.

7. To send the set request with the new value(s), SELECT the Set button at the top of the control panel.

When the Set request has been successfully completed, the Set Information list is cleared.

8.5.1 Setting Attributes

If you are allowed to set an attribute, a line is provided in the New Value column to the right of the Current Value in the attribute list. (Current Value is also provided for attributes that cannot be set; however, no lines are provided in the New Value column.) Attributes of enumeration data types that you can set have an abbreviated menu button displayed in the New Value column; press MENU on the menu button to choose the desired new value.

Creating and Managing a Link

This chapter discusses the following topics:

- Using the Console's Edit function
- Using IP Discover to create manageable links

“Manage,” in this context, means sending data and event requests to and receiving event and trap reports from. A link is an SNMP object in the category “connection.”

The easiest way to create a connection is through the IP Discover Tool, the use of which is described in “Part 2: Reference.” If you need to create only a few manageable links, you should use the Console's graphical editor function, described in the following section. The use of IP Discover hides the details of link creation; the use of the Edit function requires more intervention on your part.

9.1 Using the Console's Edit Function

You create a link in the same way you create elements of other categories. Invoke the Console's graphical editor function (press Edit►Create in the Console's control area) and click SELECT in the Connection category. See Figure 9-1.

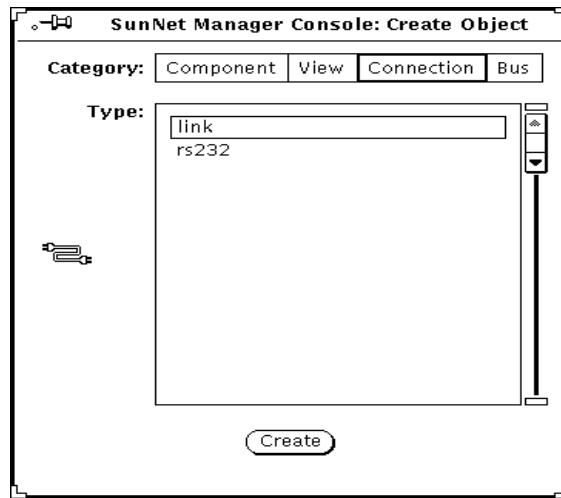


Figure 9-1 Creating a Connection Object

You can manage a connection of any type—types `link` or `rs232`, the types supplied with SNM, or connection types that you add. You can create a link between any two elements. However, realistically, the most useful links are between two components, such as two routers, or a router and a subnet.

After you click `SELECT` in the `Create` button in the window shown in Figure 9-1, you receive a properties window for a new connection as shown in Figure 9-2.

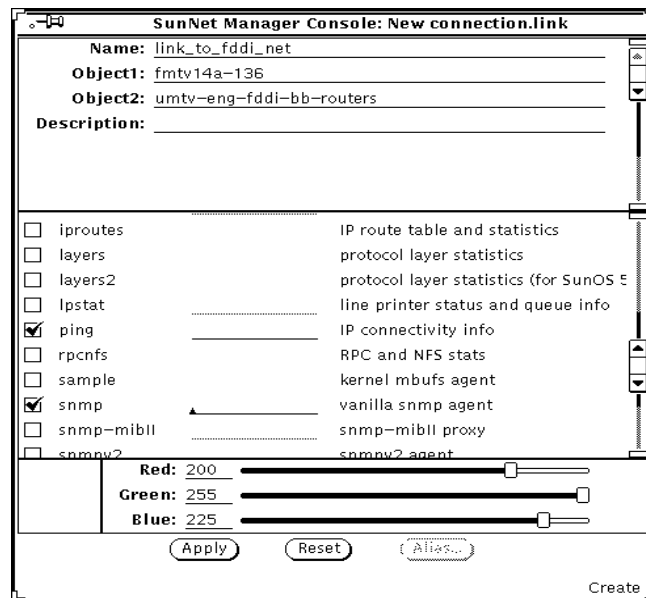


Figure 9-2 Properties for New Connection

Referring to Figure 9-2, Object1 and Object2 are the names of objects in the runtime database. SNM manages the link based on the data returned from Object1. The name of the link (in the Name field) is for your use and cannot be an existing object in the database. For a link, you should have the `ping` and, if one or both objects support it, the `snmp` agents checked off.

After you click SELECT on apply your link object is created and is displayed in the Console as shown in Figure 9-3.

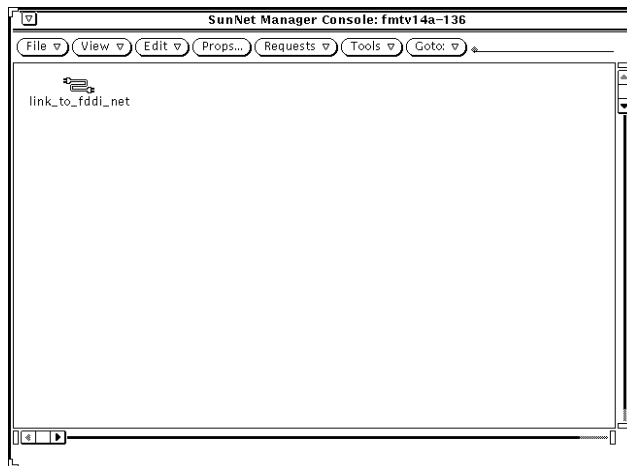


Figure 9-3 New Link in Console

This example shows the link in the view for the machine, `fmtv14a-136`, a router that is the first object (Object1) listed in the link properties. This choice of where to locate the object is arbitrary. You might locate the link in its own view, in your Home view, or the view that displays glyphs for both the router and the subnet to which it connects. In the example, we copy the link object to the view displaying the objects connected by the link. See Figure 9-4.

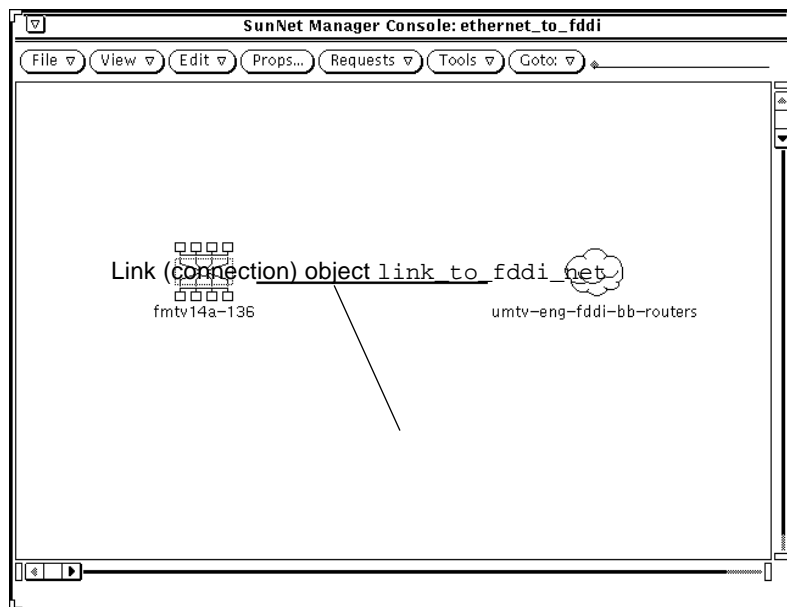


Figure 9-4 Link in View with Connected Objects

In Figure 9-4, note that the link lies between the two objects specified as Object1 and Object2 in Figure 9-2. That is, it connects the router `fmtv14a-136` to the subnet `umtv-eng-fddi-bb-routers`.

Independent of creating the object `link_to_fddi_net` in the Console, if you are creating the link object “by hand”, you must add an entry to the `linkmap` file to identify the new link. The `linkmap` file, described in a man page supplied with SNM, is stored by default in the following directory:

- `/var/adm/snm` for Solaris 1.x
- `/var/opt/SUNWconn/snm` for Solaris 2.x

For the link object `link_to_fddi_net` we create the following entry in the `linkmap` file:

<code>fmtv14a-136</code>	<code>2</code>	<code>link_to_fddi_net</code>
--------------------------	----------------	-------------------------------

The first field in this entry (`fmtv14a-136`) corresponds to `Object1` in the link properties sheet. As with `Object1` in the properties sheet, SNM manages the link based on data returned by the element in this first field. The third field (`link_to_fddi_net`) corresponds to the `Name` field in the properties sheet. The second field (`2`) is the interface on the machine specified in the first field. This is the interface over which the connection exists to the object at the remote end of the link. We obtained this number from the IP Discover Tool, so strictly speaking, we have “cheated” in this example. However, we later describe how to use IP Discover to perform the same tasks that are described here.

At this point, the link is fully manageable. That is, you can send event requests to it and, if the machine identified in the `linkmap` file is running an SNMP agent, can receive traps from it.

Sending an Event Request to a link is the same as sending such a request to a component. Figure 9-5 is an example of invoking a Send Request to a link.

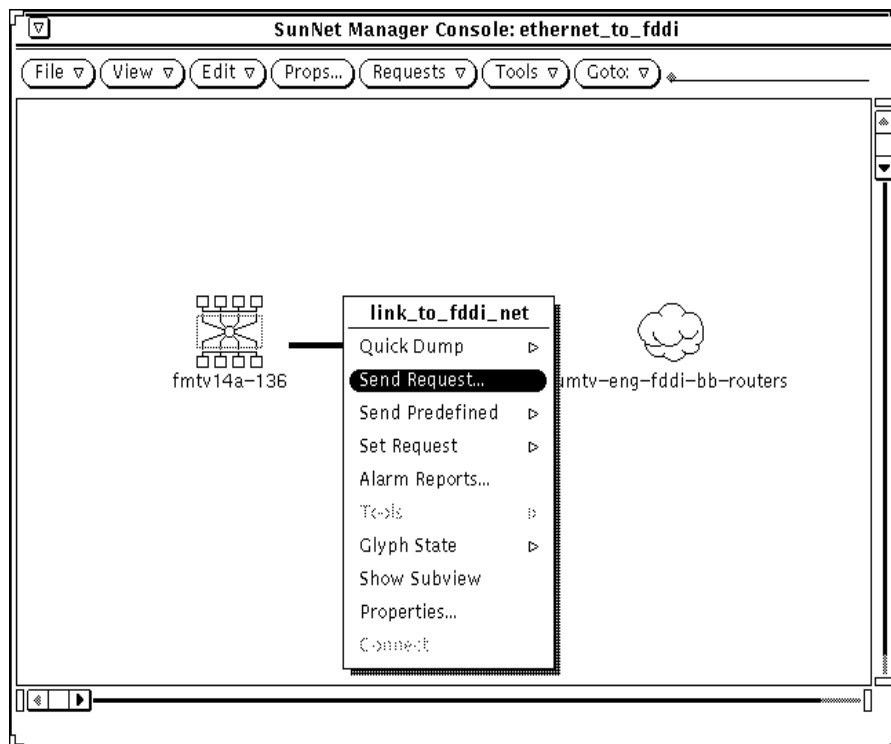


Figure 9-5 Invoking Send Request to a Link

After invoking Send Request, you receive the Request Builder window in Figure 9-6.

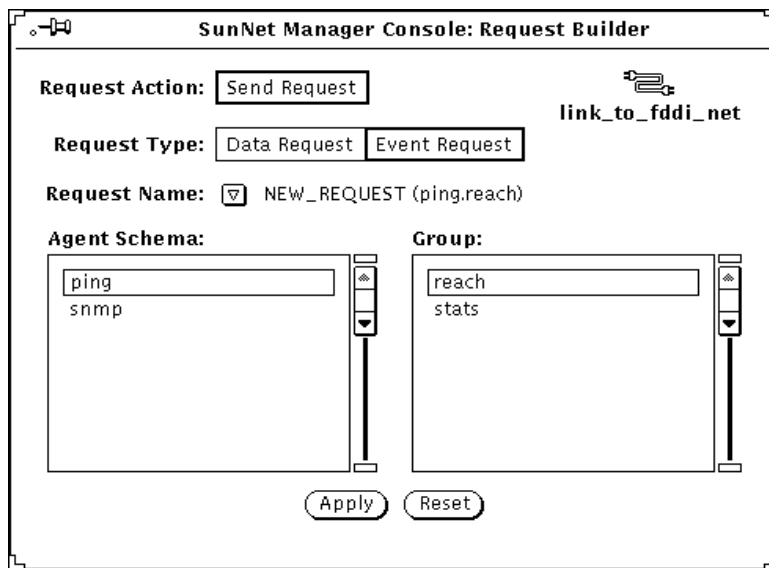


Figure 9-6 Request Builder Window

1. Click SELECT on Event Request.
2. Choose an agent schema and group.
3. Click SELECT on Apply.

You receive an Event Request properties window, such as the one in Figure 9-7.

SunNet Manager Console: Event Request (frmpk16c-076.ping.reach)

Name: ping	Attributes:
Proxy System: localhost	
Interval: 0	
Count: 0	
Key: 2	
Restart: <input type="checkbox"/>	Attribute: reachable
Send Once: <input type="checkbox"/>	Relation1: Equal To
Defer Reports: <input type="checkbox"/>	Threshold1: false
On Completion: Delete Request	Relation2: Threshold Not Set
Options:	Threshold2:
Alternate Proxy:	Priority: Low
Schedule: Off	Glyph Effect: Blink Glyph
Date Format: Month/Day/Year	Audio Effect: None
Start Date:	Audio File:
Stop Date:	Mail To:
Start Time: PM	To Program:
Stop Time: PM	Stop Request: <input type="checkbox"/>
	Start Request:

Start Hold Reset Apply Reset Delete

reachable information whether host is reachable

Figure 9-7 Event Request Properties

The Key field in the properties sheet is derived from the interface-number field in the linkmap file. The fact that this number displays correctly is one indication of a valid linkmap entry for link_to_fddi_net.

You can use the decay feature to alert you if your link goes down, then comes up again. If the link goes down (with a resultant change of glyph color to red, orange, yellow, or a color you customize) then comes back up, the glyph color changes to blue (or a color you customize) and remains that color until reset or until the link goes down again.

If the link goes down (or decays), you observe the glyph effect in the Console, just as you do for an event occurring for any element. The difference with the link management feature is that the link itself displays your specified glyph effect. When a link changes color, blinks, or dims, it indicates that one or both of the machines at either end of the link is down, or that the link medium itself is disrupted.

9.2 Using IP Discover to Create Manageable Links

The IP Discover Tool automates the process of creating manageable links and adding required entries to the linkmap file. In this example, we “discover” the same link we created by hand in the previous section.

Invoke IP Discover by selecting Tools►Discover in the Console’s control area. In the IP Discover Tool’s base window, click SELECT on Configuration Options. In the IP Discover Configuration window, to find manageable links, click SELECT on Yes in Add Object Connections. The window shown in Figure 9-8 has Add Object Connections selected.

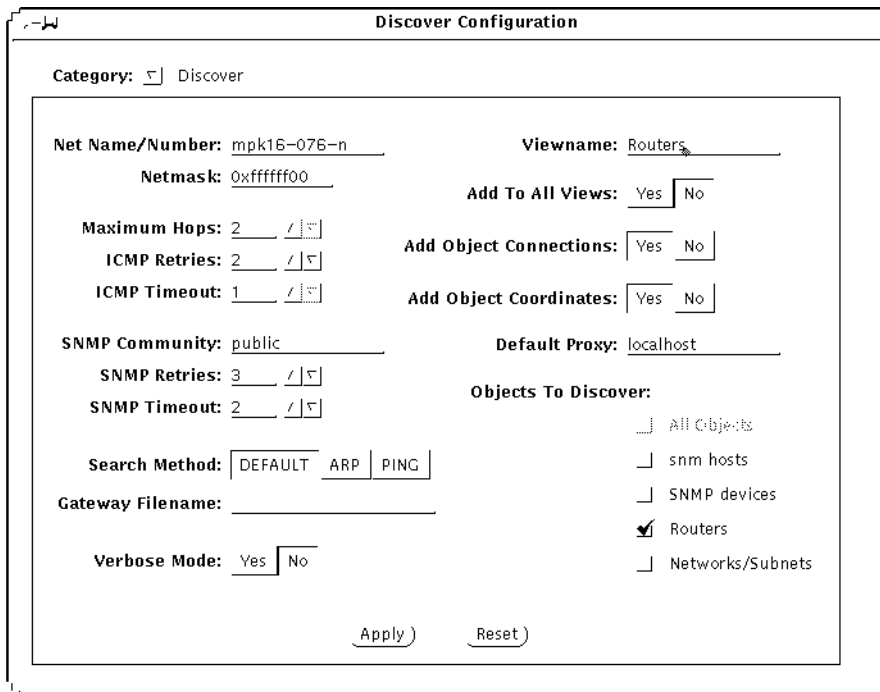


Figure 9-8 Discover Configuration Window

In addition to the Add Object Connections selection, the non-default selections in the window shown in Figure 9-8 are: Maximum Hops (changed from 0 to 1), Add To All Views (changed from Yes to No), Routers (only) for Objects to Discover. A Viewname of “routers” is specified.

Starting with version 2.3, you can choose the ARP or Ping search method in addition to the default method. The default method is a combination of ARP and Ping. See “Part 2: Reference” for descriptions of these methods.

The effect of the selections shown in Figure 9-8 is to tell IP Discover, “Find only routers 0 or 1 hops away and find all links between those routers. Add all discovered elements to a view named ‘routers’.”

After clicking SELECT on Apply and starting the IP Discover function, you receive a view similar to the one in Figure 9-9.

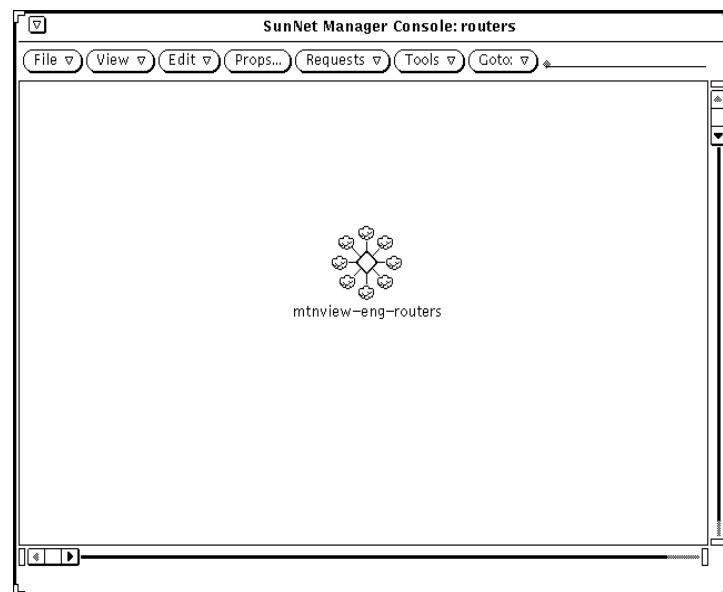


Figure 9-9 Network Icon for View “Routers”

Double-click on the network icon in Figure 9-9 to receive the sub-view shown Figure 9-10.

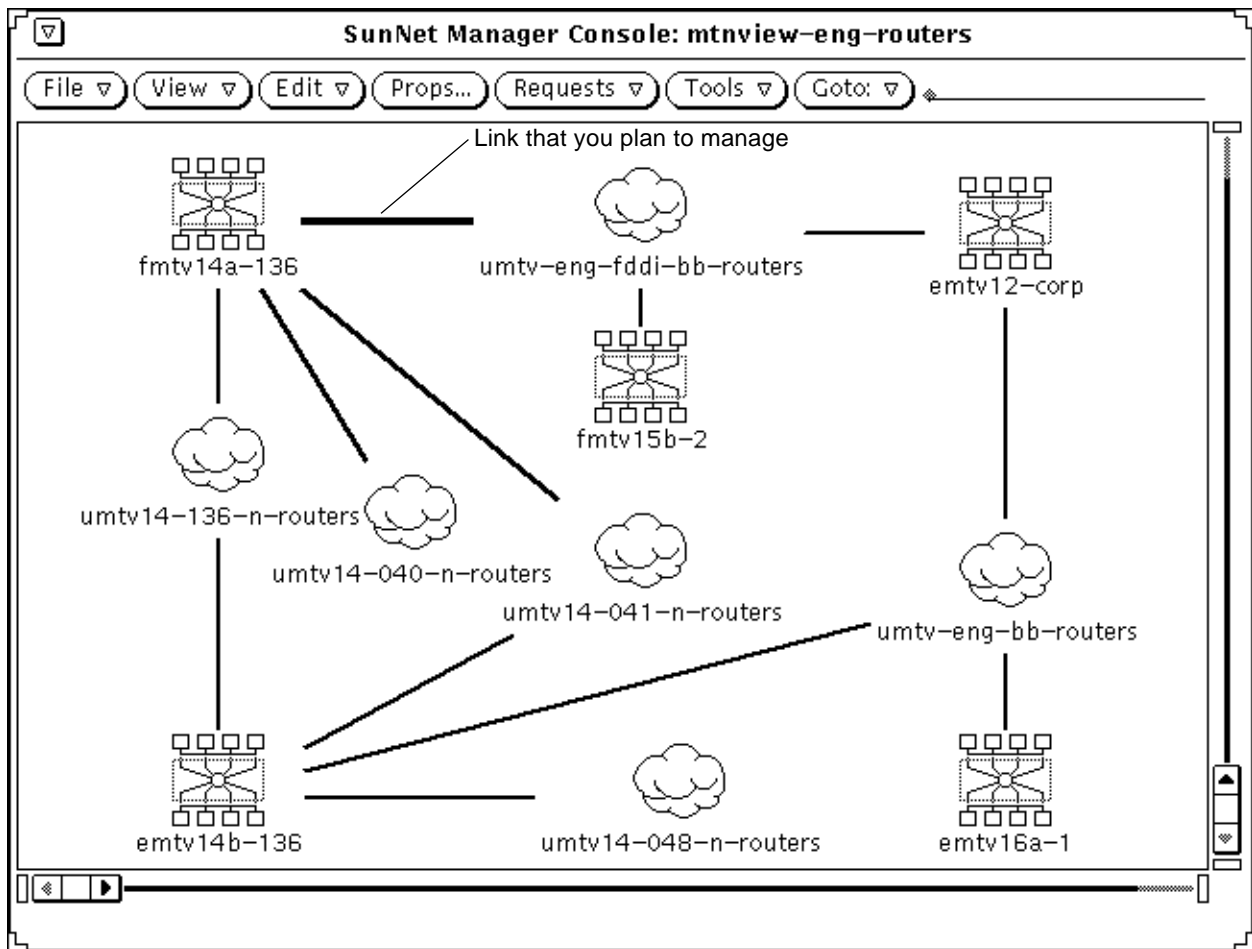


Figure 9-10 View Created by Discover

A difference between the link you created yourself in the previous section and the link created by IP Discover is the name. The IP Discover Tool creates a name by combining the names of the two linked objects and appending the string link. You can illustrate this by pulling down the glyph menu for the link glyph, as shown Figure 9-11.

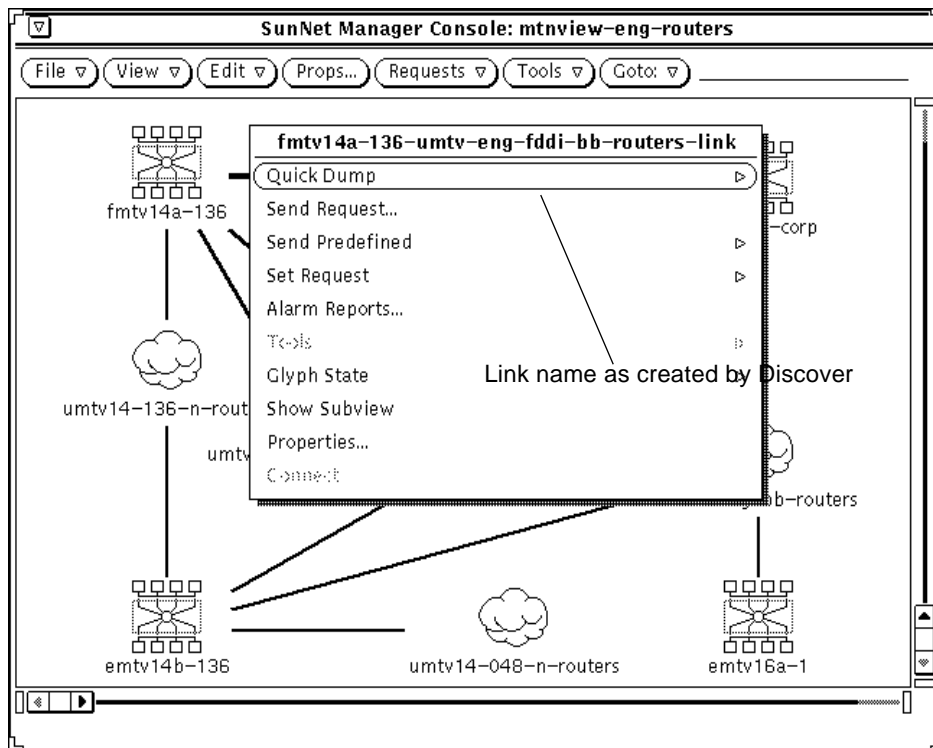


Figure 9-11 Link Name as Created by IP Discover

Figure 9-12 shows the properties of the “discovered” link. Note that, except for the name, these properties are identical to the properties for the hand-created link, shown in Figure 9-2 on page 9-3.

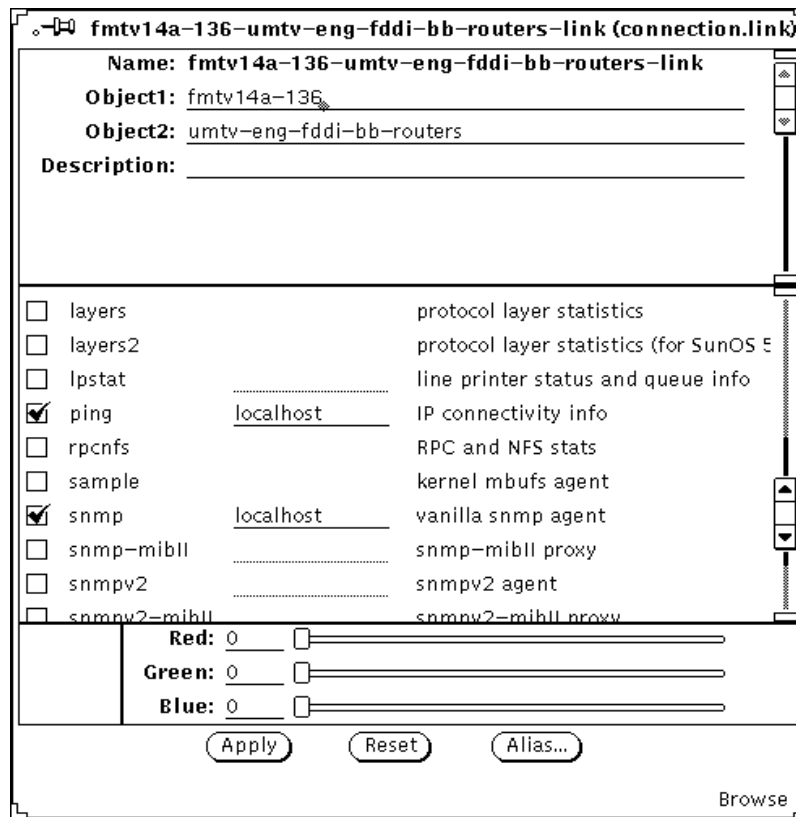


Figure 9-12 Properties of Discovered Link

As with the hand-created link, SNM manages the link based on data returned from the element in the Object1 field. Note that localhost is the default for proxy hosts, so the specification of localhost for the ping and snmp agents is equivalent to the empty proxy fields for ping and snmp shown in Figure 9-2 on page 9-3.

The decision to invoke the Add Object Connections feature in the Discover Tool should be a considered one. While the automatic creation of manageable links might be desirable, there are these disadvantages to consider:

-
- You lengthen the amount of time IP Discover takes to find elements and build your database. The time required by the monitor function to detect changes in your database is also lengthened proportionately.
 - If your network consists of a set of LANs, you create many links that have little or no value from a network management point of view. With Add Object Connections turned on, the IP Discover Tool creates links between each machine on a subnet and that subnet. In effect, you have a link that maps to the machine's network interface and its transceiver cable (in the case of Ethernet). This is of no more value than having an element for the machine (or interface) itself.
 - Following from the preceding item, you end up with a much larger database than you would otherwise.

Depending on the number of links you want to manage, it might be better for you to use the Console's graphical editor function, as described in the previous section, than to use IP Discover Tool's Add Object Connections feature.

This chapter discusses the following topics:

- Adding background image to current view
- Creating types of elements
- Creating a new glyph for an element type
- Modifying the tools menu for an element type
- Adding agents and glyphs
- Forwarding Novell's NMS Alarms to SunNet Manager

10.1 *Adding Background Image to Current View*

You may find it helpful to place elements in views relative to their physical locations to each other. A Console view may be superimposed upon a background image, such as that of a region map or the layout of a building.

1. Create the background with your favorite graphics tools (for example, Island Paint).

The background canvas can be any size. A size no larger than 1120 x 800 is recommended for standard resolution displays. The image should be in raster file format. Icon format is supported if the file ends with the extension `.icon`. Large backgrounds in icon format load much more slowly than raster format files.

2. In the Console window, invoke the **View▶Add Background** option. You receive an Add Background file menu, as shown in Figure 10-1.

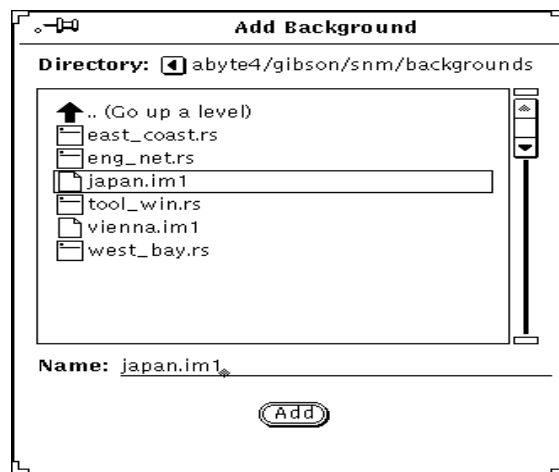


Figure 10-1 Add Background File Menu

3. **Double-click SELECT** on the file of your choice in the Add Background file menu. The background contained in the selected file is displayed in the SunNet Manager Console, as shown in Figure 10-2.

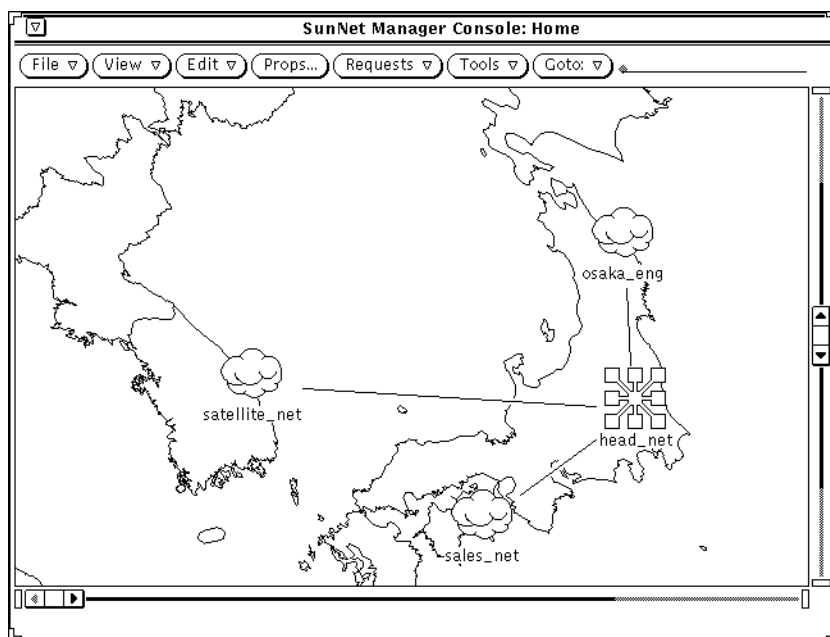


Figure 10-2 Console with New Background

After you save the runtime database to a file, the background you add to a given view is displayed each time you select that view.

To save the runtime database to an ASCII file, do the following:

1. In the Console's File button, press MENU on Save►Management Database and release MENU.

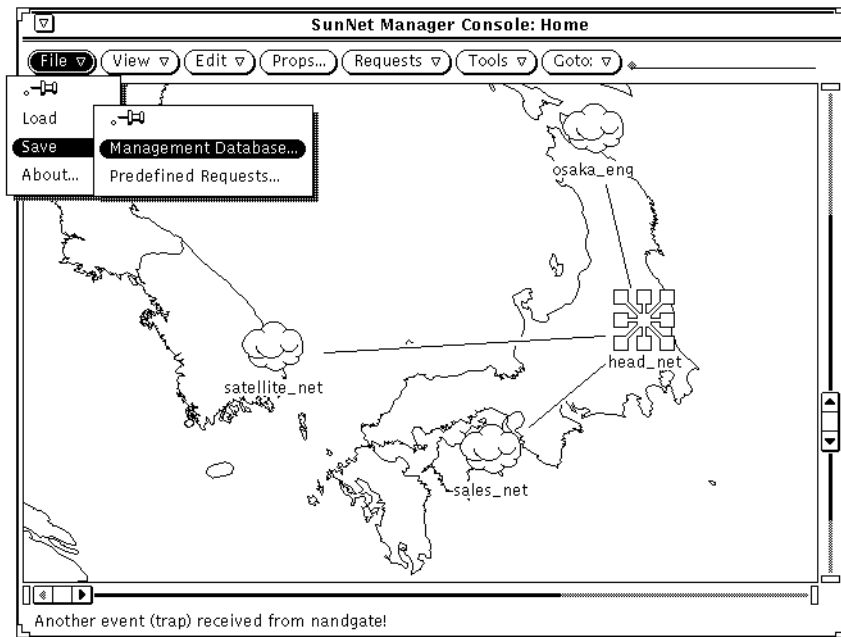


Figure 10-3 Selecting the File—Save—Management Database Menu Item

You receive the file menu shown in Figure 10-4.

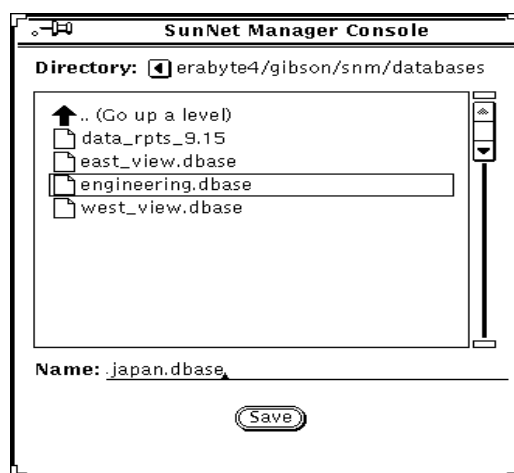


Figure 10-4 File—Save—Management Database Menu

In the file menu, enter or select the directory and file name of the file you want to save. Click SELECT on the Save button.

When you restart the Console with a command such as the one below, you receive the view with the background that you added.

```
hostname% snm -i $HOME/snm/databases/japan.dbase
```

Note that the pathname specified as an argument to `snm`, above, is an example.

10.1.1 Background Images

You can have up to 1024 icon or raster files defined in the runtime database; this includes background images as well as element glyphs.

In addition to changing the background image with the Console option described in this section, you can also edit the ASCII file that contains the runtime database to achieve the same effect. With a text editor, open this ASCII file. Define a `viewBackground` instance record that specifies the name of the view and the path name of the background image.

Database definitions for background images are described in “Part 2: Reference.”

By editing the ASCII file containing a database, you can add a background to a view other than the current view. However, it is generally easier to use the Console menu item to add or remove a background than it is to edit a file and load the modified database. Also, the Console option is far less prone to error than the file-editing option.

10.2 *Creating Types of Elements*

The `elements.schema` file defines many general types of elements, such as SPARCstations and SPARCservers. If necessary, you can create additional element types for your database.

- 1. With a text editor, create a file with the extension `.schema` in a directory specified by the Schema Directories category of the Console Properties►Locations window.**

The default locations for these directories are:

- `/opt/SUNWconn/snm/agents` or `/opt/SUNWconn/snm/struct` for Solaris 2.x
- `/usr/snm/agents` or `/usr/snm/struct` for Solaris 1.x

If you create a schema file in another directory, you should specify the directory in the Console Properties►Locations window. The steps for adding a path to this window are described in “Adding Agents and Glyphs” later in this Chapter.

- 2. Create a record that defines the element type and the fields that are stored in the database for the element type.**

Database definitions for element types are described in “Part 2: Reference.” Define a glyph to be used to represent the element type. You can also define the tools that will be available for this element type.

See “Creating a New Glyph for an Element Type” later in this Chapter for information on creating an icon for the element type. See “Modifying the Console Tools Menu” later in this Chapter for information on defining tools for the element type.

- 3. Load the new element type definition into the runtime database:**

In the Console's File button, press MENU on Load►Management Database and release MENU as shown in Figure 10-5.

An alternative to loading the file into the runtime database is to exit the Console (you may want to save the runtime database), then reinvoke the Console (`snm -i`). The new schema file will be automatically loaded in the new Console session.

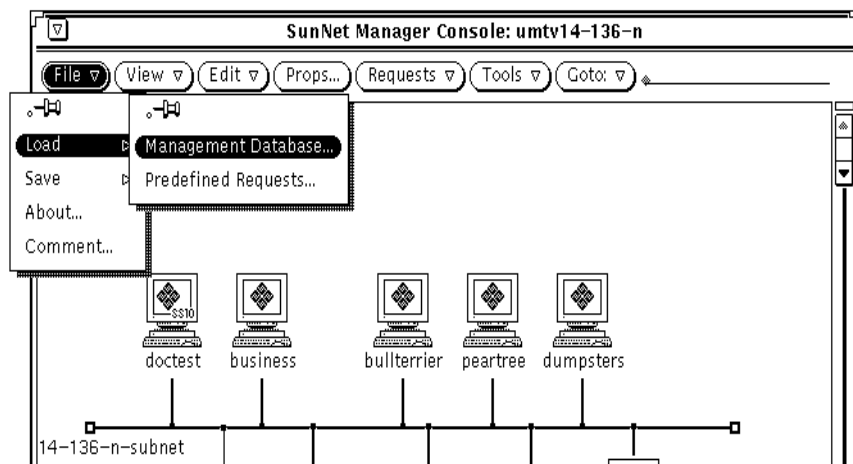


Figure 10-5 Selecting the File—Load—Management Database Menu Item

- e. In the file menu you receive (shown in Figure 10-6), enter or select the directory and file name for the schema file where the new element type is stored and click SELECT on Load.

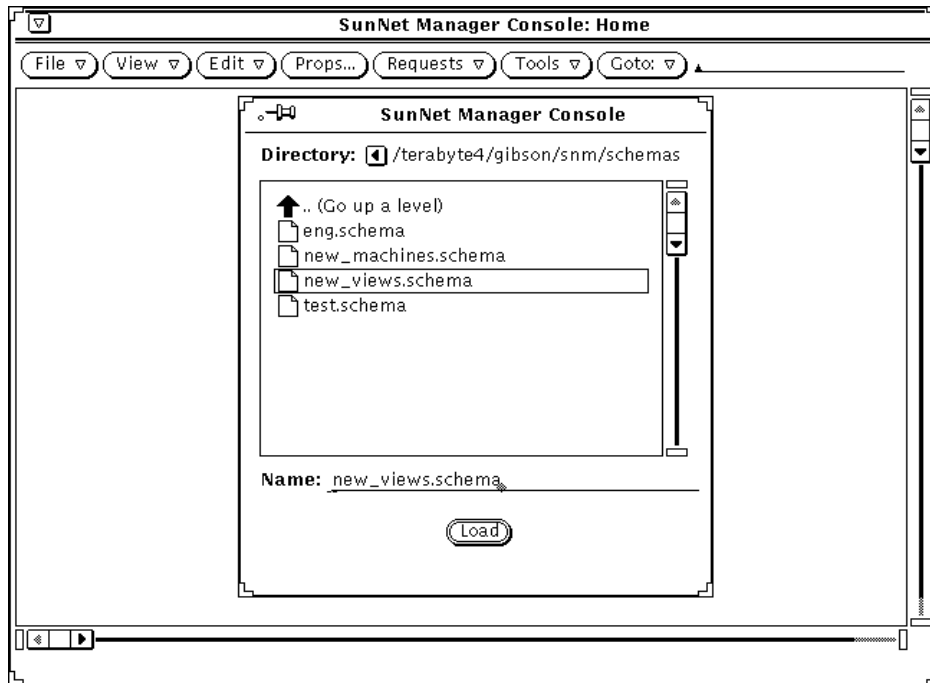


Figure 10-6 File—Load—Management Database Window

4. Create element instances of the newly loaded element type with the graphical editor.

Starting with version 2.3 a new file, `network_elements.schema`, is available which contains the Novell IPX component definition.

10.2.1 Element Categories

There are four categories of elements. You cannot create new element categories. The element categories are:

- **Component.** Includes elements such as workstations, printers, and routers.
- **View.** Includes elements such as subnets and buildings. Views can contain other views or components.

- **Bus.** Includes an Ethernet LAN segment. A bus appears as a line with movable endpoints.
- **Connection.** Includes links and RS-232 lines. A connection connects two other elements and appears as a line between the two elements.

Element types are defined in a record that specifies the element category and the name of the element type. Refer to “Part 2: Reference” for more information about the element type record definition.

10.3 Creating a New Glyph for an Element Type

Each element type defined in the `elements.schema` file is represented by a glyph. The `elements.schema` file is normally located in the following directory:

- `/opt/SUNWconn/snm/struct` for Solaris 2.x
- `/usr/snm/struct` for Solaris 1.x

You can create your own glyphs to represent an element type.

1. **Save the runtime database to an ASCII file.**
 - a. **In the Console’s File button, press MENU on Save►Management Database and release MENU. (See Figure 10-5.)**
 - b. **In the file menu, enter or select the directory and file name of the file you want to save. (See Figure 10-6.)**
 - c. **Click SELECT on the Save button.**
2. **Create an icon using the OpenWindows Icon Editor or your favorite utility. The icon file name should be `<iconname>.icon`.**

Icons shipped with SunNet Manager are 32 by 32 pixels in size. Icons to be used with the Console can be 32, 48, or 64 pixels wide and any reasonable height.
3. **You can also define a second icon—the icon mask—to be used as a stencil when coloring or inverting the icon.**

This step is optional. The icon mask should be black for bits that you wish colored or inverted. The icon mask should have the path name `iconname.iconmask`. If you do not define an icon mask, a rectangular icon is assumed.

4. In the element's schema file, define an `elementGlyph` instance specifying the component or view and its icon path name.

If the icon path name begins with a slash (/), it is treated as an absolute path. Otherwise, the path name is relative to the directories specified in the Console Properties window. Refer to "Part 2: Reference" for more information.

5. Restart the Console with the `-i` option and specify the ASCII file in which the runtime database was saved.

Note – You can have up to 1024 icon or raster files defined in the runtime database; this includes background images as well as element glyphs.

10.4 *Modifying the Console Tools Menu*

Starting with version 2.3, in addition to being able to add new tools to the Tools Menu, you can create cascaded submenus and associate commands with different menu items.

1. In the Console window, press MENU in Tools►Customize and release MENU. You receive the window in Figure 10-7.

2. Click SELECT on the Menu check box.

3. Enter the name of the Menu or Tool you are adding.

1. In the Console window press MENU in Tools►Customize and release MENU. You receive the window in Figure 10-7.

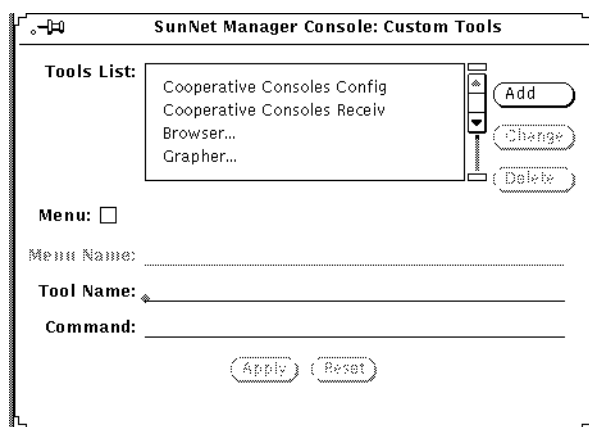


Figure 10-7 Tools—Customize Window

2. **Add a new entry or modify an entry.**
 - a. **To add an entry enter the tool name in the Tool Name line, as you want it to appear in the Console Tools menu.**
 - b. **Enter the tool's complete pathname in the Command line.**
 - c. **Click SELECT on the Add button to add your entry to the Tools List.**
 - d. **Click SELECT on Apply for your change to take effect.**
3. **To modify an entry in the Tools List, click SELECT on the entry.**

The tool's name appears in the Tool Name line as it is displayed in the Console Tools menu. The tool's full pathname appears in the command line.

 - a. **Make any edits you want and click SELECT on Change.**
 - b. **Click SELECT on Apply to make the change take effect.**
 - c. **Click SELECT on Reset to restore the menu to the state it was in when you first invoked the Customize option.**

Any changes you make to the Console's Tool menu remain in effect until the next time you invoke `snm` with the `-i` argument. The contents of the Tool menu are part of the runtime database. As such, they are saved when you invoke `File>Save>Management Database` to save the database to an ASCII file.

10.5 *Modifying the Tools Menu for an Element Type*

A SunOS command can be defined to appear in the Tools menu of an element type.

1. **Save the runtime database to an ASCII file.**
 - a. **In the Console's File button, press MENU on Save►Management Database and release MENU. (See Figure 10-3.)**
 - b. **In the file menu, enter or select the directory and file name of the file you want to save. (See Figure 10-4.)**
 - c. **Click SELECT on the Save button.**
2. **Using a text editor, open the .schema file that contains the element type for which you are adding or modifying a user command.**

If you are adding or modifying Tools menu options for an element type that is supplied with SunNet Manager, open the `elements.schema` file. The `elements.schema` file is normally located in the following directory:

 - `/opt/SUNWconn/snm/struct` for Solaris 2.x
 - `/usr/snm/struct` for Solaris 1.x

If you are adding or modifying Tools menu options for an element type that you have created previously, open the appropriate .schema file.
3. **Restart the Console with the `-i` option and specify the ASCII file in which the runtime database was saved.**

10.6 *Adding Agents and Glyphs*

See the Section, "Creating a New Glyph for an Element Type" in this Chapter for instructions on creating new glyphs.

You add agents and glyphs by appending the paths to the directories that contain them to the Schema Directories and Icon Directories items in the Console's Properties►Locations window. These might be agents and glyphs shipped with another Sun product, such as SunLink X.25, or agents and glyphs provided by a party other than Sun.

1. **In the Console window, click SELECT on Props.**

2. In the Properties window, press MENU on the Category abbreviated menu button and release MENU over Locations. You receive the window in Figure 10-8.

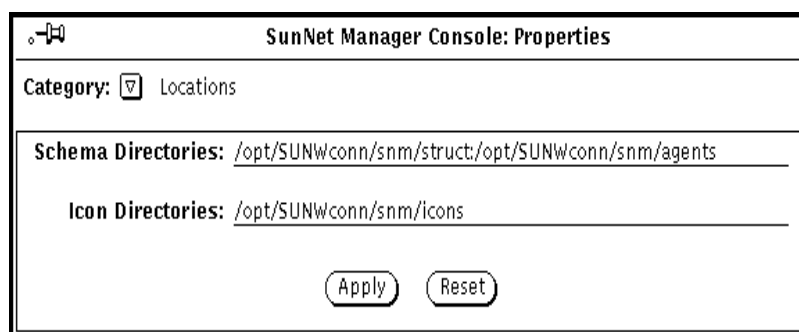


Figure 10-8 Properties/Locations Window

3. To add a path, append a colon to the last path and enter the path where your agents or icons are stored and then click SELECT on Apply. Following this, you must quit and reinvoke the Console, to make the new agents and icons available.

An alternative to the procedure described here is to simply copy the agents and glyphs (after ensuring there are no same-name collisions) to the directories already specified in the Properties►Locations window. This alternative is acceptable if you do not want to keep the SunNet Manager-supplied agents and glyphs separate from agents and glyphs you obtain from other sources. With either alternative, all agents and glyphs will be available to you whenever you create a new network element, or modify the properties of an existing network element.

This chapter covers the following topics:

- Authentication
- Access Control
- The Security Algorithm
- Conferring Right-of-Access

SunNet Manager provides an optional security mechanism for restricting access to agent services. As delivered, any manager may request data and/or event reporting from any SunNet Manager agent. For each agent, you can specify read-security for data and event report requests and write security for Set requests. This section describes how to customize security for your SunNet Manager environment.

SunNet Manager security uses and builds upon the “secure networking features” introduced in SunOS 4.x. These security features are based on DES encryption and public key cryptography. If you are installing this product on Solaris 1.x, we recommend that you first read the “Secure Networking” chapter in the *Security Features Guide* before reading the rest of this chapter. If you are installing for Solaris 2.x, you should first read *SunOS 5.2 Administering Security, Performance, and Accounting* before continuing with this chapter.

SunNet Manager implements security by giving an agent the option to authenticate each network management request before acting on it. The two elements of security are:

- Authentication—is the requestor really who he claims to be?
- Access control—does the requestor have sufficient right-of-access to make this request?

11.1 Authentication

SunNet Manager uses the RPC protocol DES authentication feature to prove the identity of the user making the request. This guarantees that the request is coming from a genuine source and makes it computationally infeasible for anyone to impersonate that source.

11.2 Access Control

Once the requestor's identity is authenticated, SunNet Manager verifies that the requestor has the required right-of-access. SunNet Manager confers right-of-access based on the "netgroups" feature of Sun OS 4.x, which was carried forward to SunOS 5.x (Solaris 2.x). Consult the following:

- The SunOS 5.2 manual *Administering TCP/IP and UUCP* (for Solaris 2.x installations),
- The `netgroup(5)` man page, and
- "Setting up Network Software" in the *System and Network Administration Guide* (for Solaris 1.1 installations).

Access rights are administered by granting membership in one of the five network security groups. These groups are named:

- `netmgt_security_one`
- `netmgt_security_two`
- `netmgt_security_three`
- `netmgt_security_four`
- `netmgt_security_five`

Access control for members of this group works as follows:

The Agent Library has a global variable, Network Management Security Level, that can have a value between 0 and 5. The value 5 is the most secure and will allow only members of the network group `netmgt_security_five` access to network management functions served by this agent library. All other requestors will be refused. The value 4 will allow access to members of groups

`netmgt_security_four` and `netmgt_security_five` and so on. Lower values of the Network Management Security Level variable allow access to members of more and more network groups and the value 0 provides no access control at all.

For the security scheme to work, both the manager's and the agent's machine (and their respective NIS/NIS+ master server machines) must be running under SunOS 4.0 or later, and the `keyserv` daemon must have been started at boot-time in one of the following ways:

- By the `/etc/rc2.d/S71rpc` script, if the installation is in a Solaris 2.x environment
- From the `/etc/rc.local` file, if installed in a SunOS 4.x environment.

11.3 The Security Algorithm

The following algorithm is used by the agent library to verify the authenticity and the access rights associated with every incoming request.

1. Since all security and authentication depends upon NIS/NIS+, first check if NIS/NIS+ is up and running. If NIS/NIS+ is down, reject the request indicating that NIS/NIS+ is not running.
2. Check Network Management Security Level to ascertain the level of security being requested. If this value is zero, bypass the following checks and allow this request through without authentication.

Note – UNIX-style authentication is not supported.

3. Verify that the request has a DES style authenticator. If not, the request is refused, indicating too weak an authentication.
4. Verify that a domain specific user-id and password table entry exists for the network name supplied in the DES authenticator contained in the request. If not, the request is refused.
5. Finally, confirm membership of the appropriate network groups based on the value of Network Management Security Level (for a value of 1, check groups 1 through 5; for a value of 2, check groups 2 through 5; etc). Deny the request if membership is not confirmed.

If the request passes the above tests, it is considered successfully authenticated, and processing of the request proceeds as normal.

11.4 *Conferring Right-of-Access*

Now that you understand how security works, let's look at how to confer right-of-access for your agent.

1. Create up to five network groups for the network named `netmgt_security_one`, `netmgt_security_two`, `netmgt_security_three`, `netmgt_security_four`, and `netmgt_security_five`.
2. Include in these groups the user-names of the system administrators who have permission to run SunNet Manager. This grouping defines a hierarchy of abilities for the administrators. Those administrators who are members of the `netmgt_security_five` group have maximum privilege and can send requests to any agent. Members of the `netmgt_security_four` group can send requests to agents with security level 4, 3, 2, 1 or 0. Members of the last group, `netmgt_security_one`, can send requests to agents with security level 1 or 0.
3. For each administrator, create a new public key for the administrator's user-name using `newkey(8)`.
4. For each host on which you have agents installed, create a new public key for the super-user at the host using `newkey(8)`.
5. Decide the level of read-security and write-security you wish to assign to your agents and set the security levels associated with the agent name in the SunNet Manager configuration file `snm.conf` to values between 0 (no security) and 5 (maximum security) on each system where agents are installed.

An example entry is shown below:

```
na.snmp      2  4
```

This sets the SNMP proxy agent read–security level to 2 and write–security level to 4. A user must be in at least the `netmgt_security_two` network group to request SNMP data or event reports and in at least the `netmgt_security_four` group to invoke SNMP Set requests.

If the `snm.conf` file does not contain an entry for an agent, the agent’s read *and* write security levels default to 0—no security checking. If the entry for an agent contains only one number, the agent’s read *and* write security levels are set to that number.

The Network Layout Assistant (NLA) provides you with clear, readable views of your networks and helpful tools for documenting and navigating through them.

12.1 Who Should Use NLA?

- Anyone whose network grows beyond a few devices; it is far easier to manage your maps with automated layout
- Anyone who has network topologies that change often.
- Anyone who needs high quality network maps — for example, for presentations, reports, and demonstrations

12.2 What Does NLA Do?

- NLA reads the network information in your SunNet Manager database, and automatically places devices and connections in a meaningful arrangement in the Console window. There is no need to hand-position each device as you build a logical view of your internetwork.
- NLA shows you where you are. It is often difficult to see an entire view of your network with the Console since there is no “zooming” feature. The NLA overview window provides a high-level view of the Console and a marquee indicating the portion of the network you are currently viewing in the Console window. The marquee mirrors your moves as you scroll through the Console, thereby helping you navigate your network.

- It provides you with choices. Since internets can be comprised of a variety of topologies, NLA provides three unique layout styles: Hierarchical, Circular and Symmetric. Experiment with each style to find the best one for you.
- It lets you customize your layouts. Although each layout style has preset defaults, you can tailor the effect through OpenWindows controls.
- It lets you print your networks. Using the Print option, you can generate color PostScript files of your network views and include them in page composition programs or send them directly to a PostScript printer/plotter.

12.3 What the Network Layout Assistant Does Not Do

- The Network Layout Assistant is not a Discover tool. NLA cannot discover your network devices. Specifically, NLA does not populate the SunNet Manager database with connection information. However, NLA will accurately lay out the device and connection information stored in the SunNet Manager database.
- The Network Layout Assistant does not provide a physical map or geographical layout of your network. It provides logical arrangement of your devices based on their connectivity and does not represent the actual distance between devices or physical location of devices.
- The Network Layout Assistant does not troubleshoot network errors. However, it can help you resolve and avoid problems through its clear presentation of complicated network maps.

12.3.1 Available with Domain Manager

The Network Layout Assistant is available only with the Domain Manager product, not with Site/SunNet Manager.

12.4 Starting the SunNet Manager Console

Access NLA through three new menu items under the submenu NLA in the SunNet Manager Console Tools menu: **Layout...**, **Overview...**, and **Print...**

If this is the first time you are using NLA tools but you have previously run SunNet Manager, you must first clear the current Console run-time database:

- To save the current runtime database, start the Console and use the File: Save menu to save it to a named file; then quit the Console.
- To clear the runtime database, start the Console with the `-i` option:

```
% /usr/snm/bin/snm -i &
```

If you have never used the Network Layout Assistant, or have never run the SunNet Manager Console, you can start the Console by entering:

```
% /usr/snm/bin/snm &
```

More information on starting the SunNet Manager Console can be found under “Starting the Console” in the “Creating and Modifying the Management Database” chapter. Issuing the `snm` command causes an empty Console window to appear as shown in Figure 12-1

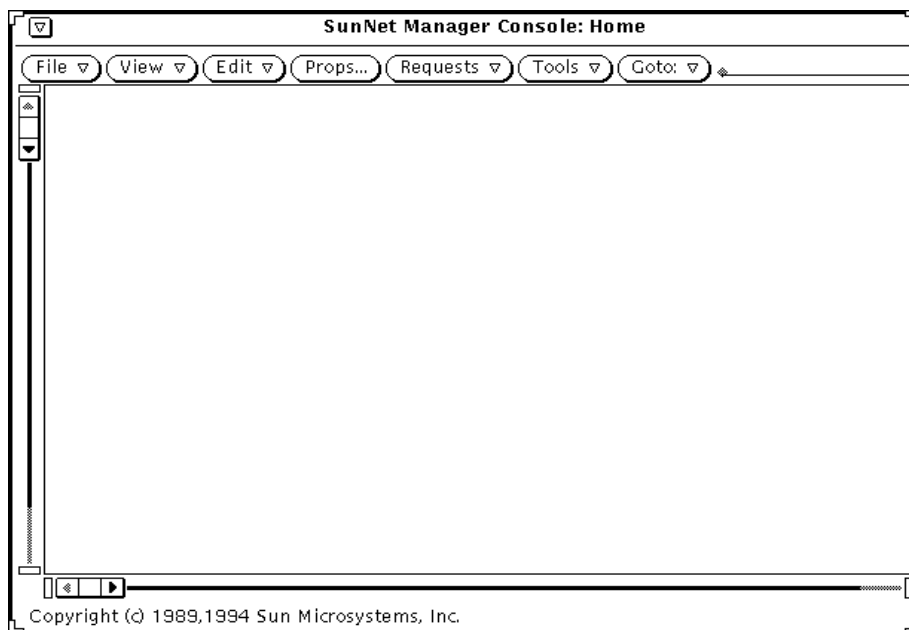
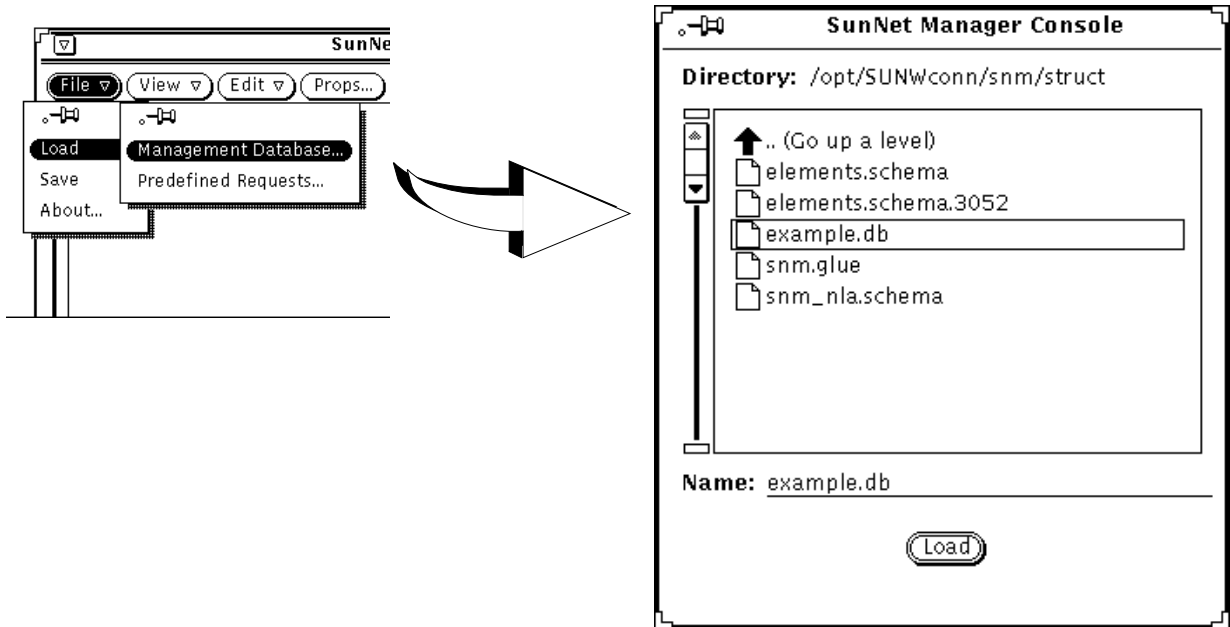


Figure 12-1 Console Window

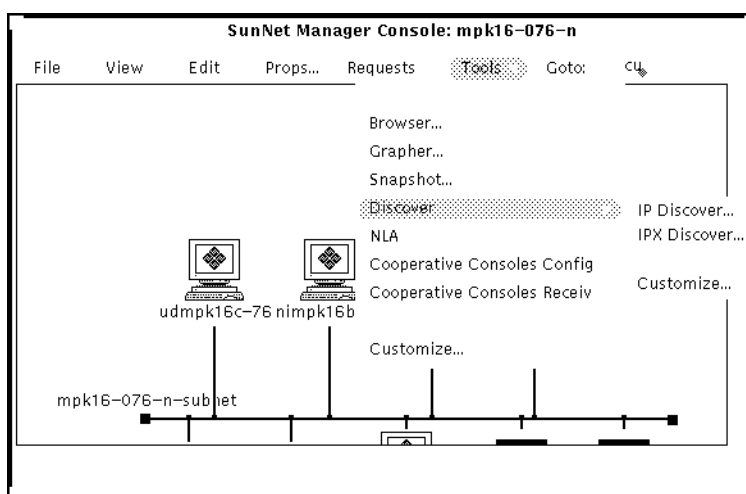
12.5 Creating the SunNet Manager Console Database

You can populate the SunNet Manager Console runtime database with your network information in a number of ways as described in other chapters. These are summarized as follows:

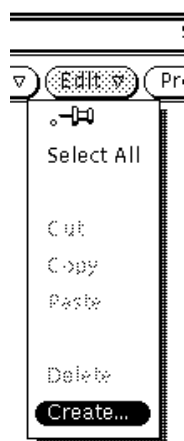
1. If you have previously built your network using the Console and have saved the database to a file using **File>Save**, you can populate the SunNet Manager Console by selecting the **File>Load** menu item. Then reload the database using the pop-up window:



2. If you do not have a saved database to reload, you can use the **Tools>IP Discover** menu option to find all the devices in your network and populate the Console database:



3. Instead of using the IP Discover tool, you can use the Console **Edit>Create** menu to create new views and devices as described in the section entitled “Creating Elements” in the “Creating and Modifying the Management Database” chapter.

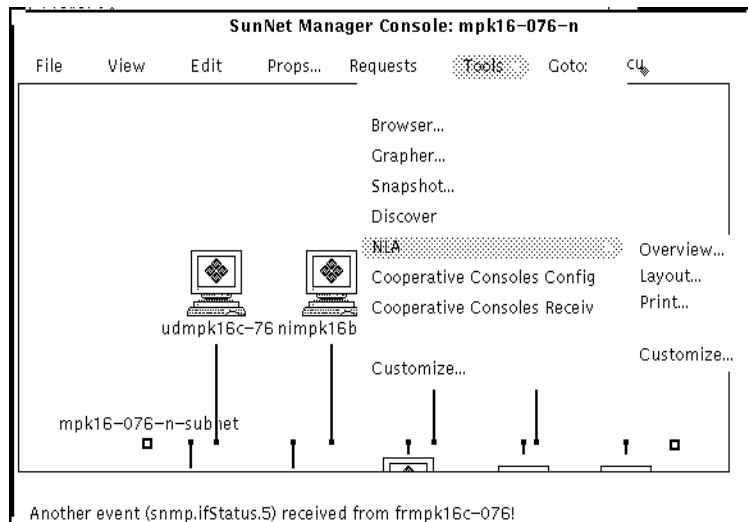


4. After adding the elements to the Console, add connections between the glyphs as necessary as shown in the “Connecting Elements” section in the “Creating and Modifying the Management Database” chapter.

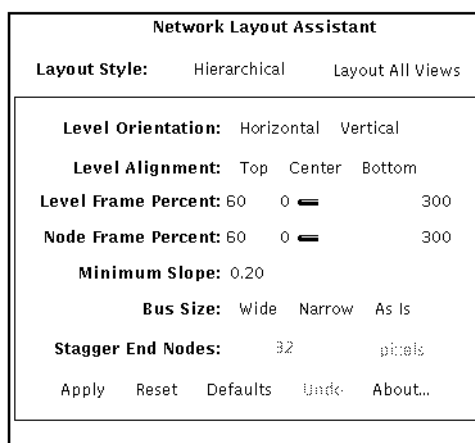
If you used IP Discover to populate your database, you might want to use the Edit>Copy and Edit>Paste options to copy elements from multiple “discovered” views into one logical view. This new logical view will help you understand the logical interconnection of your networks.

12.6 Using the Layout... Option

To bring up the Network Layout Assistant main window, select the Tools>NLA>Layout menu option on the SunNet Manager Console:



The first time you start the Network Layout Assistant, the pop up window will show the default settings:



1.

To lay out your network using the default values, click the Apply button. NLA will position the elements in the view considering their connections. The defaults lay out the current view using the Hierarchical style. To lay out all views of your network at once, Layout All Views►Apply. To use a different type of layout, use the Layout Style menu as described in the section on “Choosing a Layout Style.”

12.6.1 Main Window Controls

Use the NLA main window to choose your layout style, determine which views it will affect, and apply it to the Console contents. Depending on the style, style-specific panels offer various options that further affect the overall glyph positioning.

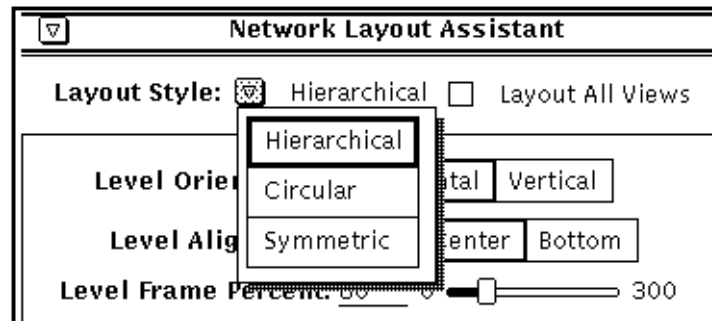
These controls are present on all the layout style windows:

- **Layout Style** menu offers a choice of three layout styles: Hierarchical, Circular or Symmetric.
- **Layout All Views** check box determines whether the Network Layout Assistant lays out all views of your network at once, or just the current view.

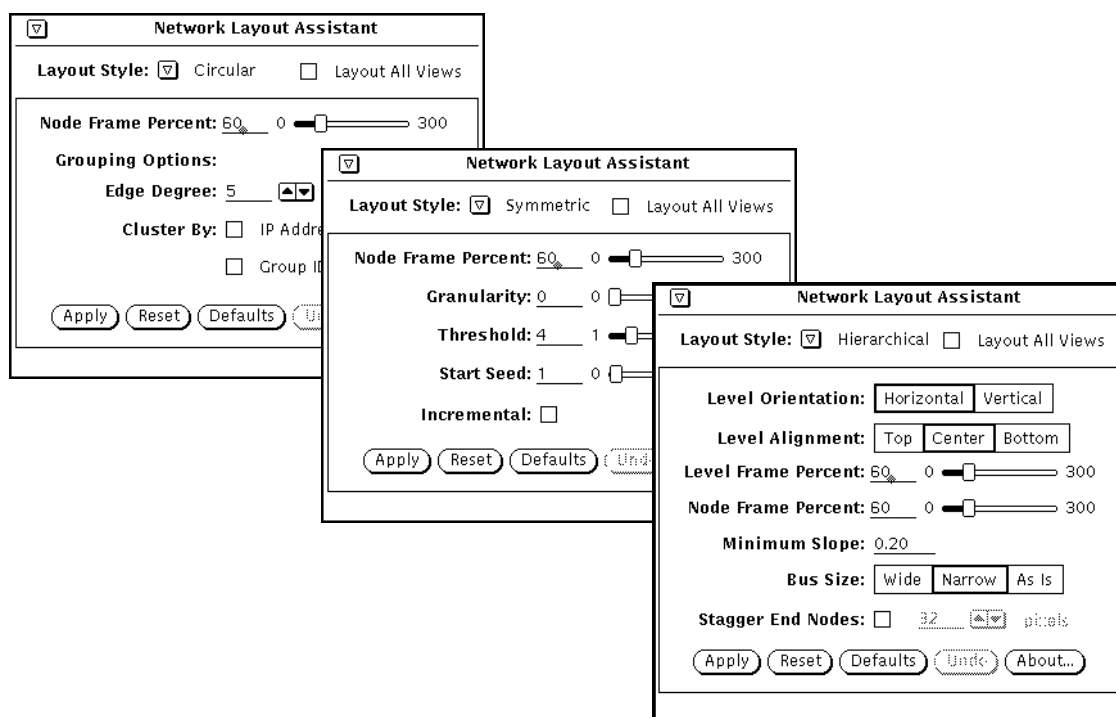
- **Apply** button causes the settings in the Network Layout Assistant main window to take effect and a layout to be executed.
- **Reset** button returns the current style's settings to those before you last pressed Apply.
- **Defaults** button resets the current style's settings.
- **Undo** button returns the network elements to where they were located before you pressed Apply.

12.6.1.1 *Choosing a Layout Style*

You choose a layout style by pulling down the Layout Style menu and selecting one from the list:



Selecting from the Layout Style menu will display one of three panels from which you can set the layout style specific tailoring options:



Layout style is primarily a matter of personal preference, and your preference may change as your network changes. Experiment with the different layout styles and settings to see which work best for you.

The layout style-specific tailoring options are described in the section, “Tailoring Your Layouts.” All three layout styles have sophisticated real-time layout capability and support the following features:

- Variable spacing between and around elements.
- Line crossing minimization.
- Separate placement of independent internetworks to help you isolate problems.
- Name-sorted tiling of independent elements.

If there are no connections between your glyphs, each layout style will look virtually the same. When there are no connections, the Network Layout Assistant provides a sort-by-name tiling of glyphs similar to the SunNet Manager IP Discover tool's automatic positioning feature.

12.6.2 Troubleshooting

- Executing a layout buries the buses underneath the connections.

Explanation: The SunNet Manager Console draws bus elements underneath the connections. Click on a bus to bring it to the front.

12.7 Using the Overview... Option

The Network Layout Assistant overview window displays a high-level view of the network contained in the Console window. The overview window helps you maintain your perspective as you scroll around the SunNet Manager Console. A marquee in the overview window outlines the portion of the network visible in the Console.

To familiarize yourself with the overview window's capabilities, follow these steps:

1. In the Console window, select the Tools►Overview menu item, and the overview window will pop up. Depending on the size of your network, it can sometimes take up to thirty seconds for the overview window to initialize itself from the SunNet Manager database.

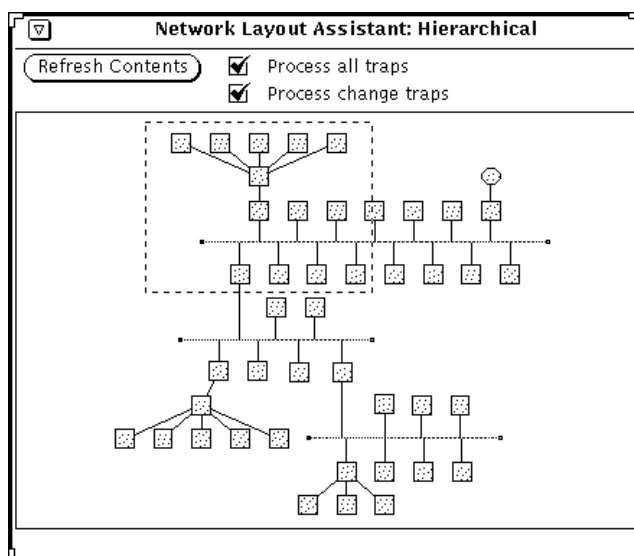


Figure 12-2 Overview Window

2. The overview window is read-only. To move its marquee, use the scroll bars in the Console. The overview window's marquee will automatically update itself to reflect the Console map's visible area.
3. As you resize the Console or scroll around it, the overview window automatically adjusted to show your current view of the network.

Note – An error message will appear when you open a view that you have not yet laid out. This message reminds you that the glyphs in that view won't be visible in the overview window until their positions are initialized either through hand positioning or Network Layout Assistant positioning.

12.7.1 Overview Window Controls

12.7.1.1 Refresh Contents Button

The overview window's network representation can sometime appear out of sync with the Console window view. This can happen, for instance, when you load a new database file using the Console's File►Load menu option. To synchronize the overview, click the Refresh Contents button to update it.

12.7.1.2 Process All Traps Check Box

To temporarily turn off the overview updates, uncheck the Auto-update check box. When unchecked, changes made to the Console will not be reflected in the overview window. To synchronize the overview window with the Console, simply reselect the check box.

12.7.1.3 Process Change Traps Check Box

The Overview window receives traps from the SunNet Manager database for actions such as cut, paste, drag, and color changes whether these actions were executed by the user or by SNM applications. Some SNM applications can generate a high frequency of database traps, causing the Overview window to use a larger portion of the CPU as it processes these traps. If you uncheck the Process Change Traps check box, the Overview window will ignore certain change traps, significantly reducing the processing requirements. whether this setting is on or off, most users will not notice a performance difference; if you experience reduced window manager performance caused by a high frequency of redraws in the Overview window, try turning this option off.

12.7.2 Shapes

The overview window emulates the SunNet Manager Console:

- Routers are represented by diamonds.
- Subnets are represented by ovals.
- Buses are represented by knob terminated lines.
- All other components and views are represented by rectangles sized proportionally to the glyph representing the component or view.

12.7.3 Color

The overview window uses the color of the network element as defined in the SunNet Manager Console.

To set the color of a Console element, use the Red/Green/Blue sliders in the element's properties sheet. Colors for window decorations such as borders and scroll bars are controlled by OpenWindows. To change them, choose the OpenWindows Desktop ► Properties... pop-up menu item.

12.7.3.1 For More Information on Setting the Color

On setting the color when creating an element:

See the section entitled "Creating Elements" in the "Creating and Modifying the Management Database" chapter.

On setting the color for an existing element:

See the section entitled "Modifying Element Properties" in the "Creating and Modifying the Management Database" chapter.

On setting a color for window decorations:

See the OpenWindows on-line tutorial or the *OpenWindows User's Guide*.

12.7.4 Troubleshooting

The glyphs and/or connections in the Console do not show up in the overview window, or do not appear to be in the correct position.

Explanation: If you have dragged elements, they may not appear in their new location because dragging does not update the position in the database. Likewise, if you have added connections between glyphs that do not have set positions, the connections may not show up in the overview window.

You can use any of the following approaches to remedy the situation:

- If you drag elements and would like the overview window to reflect your changes, switch from the current view to a different view and back again. You can switch between views by either using the Goto menu or by double-clicking on the displayed view.

- Set positions for all elements by applying a layout with the Tools►Layout menu item.
- Cutting and pasting also sets the glyph positions. Position each element manually by cutting then pasting instead of dragging.

12.7.4.1 *For More Information:*

On the difference between explicit and automatic positioning for elements:

See the section titled “Moving Elements” in Chapter 3, “Creating and Modifying the Management Database.”

On visiting a view with the Goto menu or by double-clicking an view:

See the section titled “Traversing the View Hierarchy” in Chapter 3, “Creating and Modifying the Management Database.”

12.8 *Using the Print... Option*

This section describes how to use the options on the Print window to create PostScript files of your SunNet Manager views. You can generate wall-size diagrams to improve network documentation capabilities, or generate small images to include in other documents. The printing features include:

- Single-page and multi-page output by specifying number of row and column pages. Images will be scaled to fill the number of specified pages.
- User specified page width and height settings provide arbitrary PostScript plotter size support.
- With EPSF support, network drawings may be easily embedded into documents generated by page composition applications such as FrameMaker and PageMaker.
- Gray scale and color support
- Crop marks for paper cutting
- Page labeling and numbering
- Image rotation
- DSC 3.0 support

12.9 The Print Window

To bring up the NLA print window, select the Tools>NLA>Print menu option on the SunNet Manager Console.

To print the current view using the default settings, check the printer name and click the Print button. The Network Layout Assistant will send a one-page image to the printer specified.

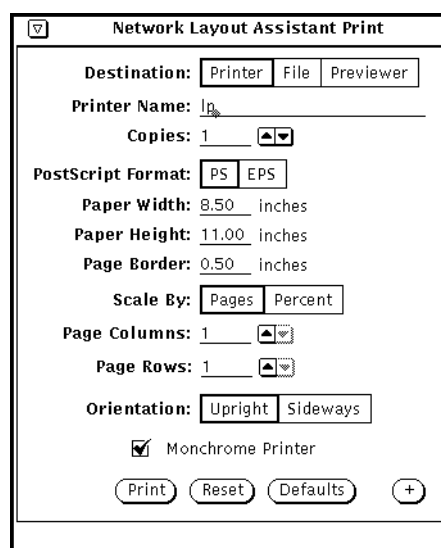


Figure 12-3 Print window with default settings

When the print button is pressed, the Network Layout Assistant will read through the current view and generate a PostScript file showing the devices and links.

Note – Unpositioned elements will not be processed by the print program. Use the layout option or hand position your devices before printing.

Note – Strip-charts in the current view will be displayed as shaded boxes

12.9.1 Primary Print Options

This section describes the options on the primary print window.

12.9.1.1 Destination

The generated PostScript can be sent directly to a printer, saved as a file, or passed to a user supplied PostScript previewer program:

- Printer

To send the output directly to a printer, select “Print” as the destination and enter your printer name in the “Printer” text field. The current view will be processed and the output routed directly to that device using `lpr` (Solaris 1.x) or `lp` (Solaris 2.x).

- File

To save the output into a file, select “File” as the destination and enter the directory and file name in the supplied text fields. In the directory field you can use the shell home directory symbol `'~'`. In the filename field you can use `$VIEW` to represent the current SunNet Manager view name. For example, to save the current view “MyNet” into PostScript file `MyNet.ps` in your home directory, specify a directory of `“~”` and a filename of `“$VIEW.ps.”`

- Previewer

To open the file in a PostScript previewer, select “Previewer” as the destination and enter the preview command line in the text field. The previewer will be launched in the background using the Unix “system” command. To represent the name of the PostScript file in the previewer command you can use the `$FILE` variable.

If no `$FILE` variable is specified in the previewer command, the filename of the temporary PostScript file will be appended to the end of the command. For example, to start page view at 72 dots per inch scaling level you would specify `“pageview -dpi 72 $FILE”` in the Command field. Note that in this case since `$FILE` is specified at the end of the command line, the same effect can be realized without including `$FILE` in the command line at all, i.e. `“pageview -dpi 72”` would be sufficient.

12.9.1.2 *Format*

The Format choice determines whether a standard single/multi-page printer ready PostScript (PS) file or an encapsulated PostScript (EPS) file is created. The EPS file can be imported into most composition programs such as FrameMaker and Illustrator, while the standard PS file can be sent directly to a printer/plotter.

12.9.1.3 *PS Output*

When generating standard printer-ready PostScript files, a number of further options are available which determine the page size and number of pages for the output file.

First you should specify the page size and page border using the Page Width, Page Height, and Page Border fields. The number of pages of this size that will be generated is determined by the “Scale To” setting:

- Scale to pages

When you select “Scale To: Pages”, entry fields are displayed where you can specify the number of page rows and page columns. To generate a single page of output enter a “1” in both fields. To split the output onto four pages (two pages by two pages), enter a “2” in both fields.

- Scale to percent

You can allow the Network Layout Assistant to set the number of pages based on the size of the current view using the “Scale To: Percent” choice. A setting of 100 percent will produce an overall image of the same size as the current view assuming there are 72 pixels per inch (1 pixel will be equal to 1 point). This overall image size is divided by the specified Page Width and Page Height to determine the actual number of page rows and page columns.

12.9.1.4 EPS Output

When generating an EPS image, the size of the image is determined by the “Image Width,” “Image Height,” “Image Border,” and the “Scale To” options.

To specify an exact image width and height, choose “Scale To: Both” and enter the width and height value in inches. To just set one dimension and let the Network Layout Assistant determine the other dimension based on the aspect ratio of the current view, choose “Scale To: Width” or “Scale To: Height” and enter either the “Image Width” or “Image Height”, respectively. For example, if you want an image 4.5 inches wide but want the height based on the current view proportions, select “Scale To: Width” and enter “4.5” in the “Image Width” field.

You can also let the Network Layout Assistant set both the width and height of the image using the “Scale To: Actual Size” option. This will generate a single EPS image of the same size as the current view assuming there are 72 pixels per inch.

The “Image Border” option allows you to specify extra white space or to display a image border/label within the specified image height/width.

12.9.2 Additional Print Options

Secondary output options can be accessed by pressing the “+” button in the lower right hand corner of the Print dialog. The Print window will open up to display a secondary panel with additional options:

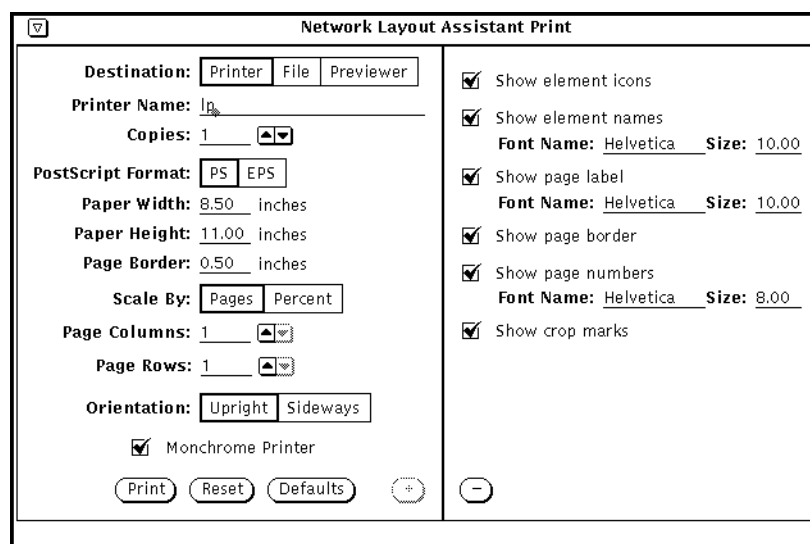


Figure 12-4 Print window with Secondary Option Panel Displayed.

These options are:

- Show element icons

If you turn this option off, the devices in the output will be represented with shaded boxes instead of including the actual bitmap. This option may be useful for producing draft output on plotters where bitmap rendering is time consuming.

- Show element names

This option determines whether the device labels will be shown on the output. The accompanying text fields are where you describe the PostScript font and point size with which to draw the device labels.

- Show page label

This option determines whether the page labels, including userid and time stamp, will be shown on the output. The accompanying text fields are where you describe the PostScript font and point size with which to draw the page label.

- Show pageborder

This option determines whether a black rectangle will be drawn encompassing the overall image.

- Show page numbers

This option determines whether the page numbers will be shown on the output. The accompanying text fields are where you describe the PostScript font and point size with which to draw the page numbers.

- Show crop marks

This option determines whether crop marks describing where to cut the paper are included on multi-page output.

Note – If Page Border/Image Border is 0.0, the page border, page label, and page numbers will not be displayed regardless of the checkbox settings in the secondary panel.

12.10 Tailoring Your Layouts

This section explains how to make adjustments to your network layout through the style specific controls in the Network Layout Assistant main window. The three layout styles – Hierarchical, Circular, and Symmetric – are each covered in their own section.

12.11 Hierarchical Layout Style

- Organizes the elements in your network into a hierarchy.
- Emphasizes tree-like subnetworks in your network.
- Is excellent for maps containing buses, because the Network Layout Assistant can stretch the bus glyphs to accommodate all of the elements directly attached to them.

Figure 12-5 shows the Network Layout Assistant main window with the Hierarchical layout style default settings.

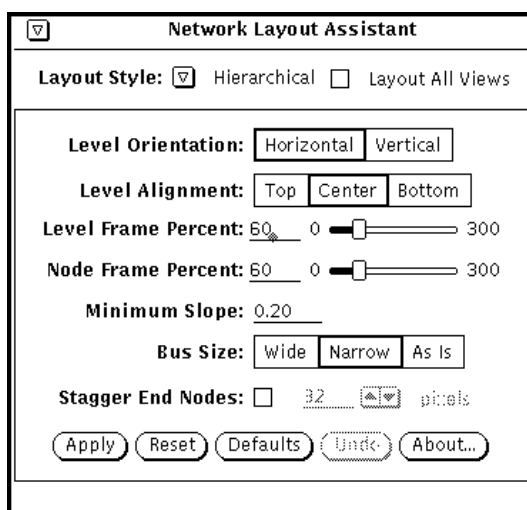


Figure 12-5 Main Window - Hierarchical Layout Style Default Settings

12.11.1 What is a Level?

A level is a grouping of nodes placed on a line. Usually a horizontal or vertical line can be drawn through the nodes that belong to a particular level. If a graph has a horizontal level orientation, each level occupies a row in the graph. If a graph has a vertical orientation, each level occupies a column in the graph.

12.11.2 Hierarchical Layout: Level Orientation

The Level Orientation panel option changes the orientation of the levels in a graph between horizontal and vertical. When the orientation is horizontal, all the nodes will line up on horizontal levels. Horizontal levels imply a top-to-bottom organization of your network in the Console. When the orientation is vertical, all the nodes will line up on vertical levels. Vertical levels cause a left-to-right organization of your network in the Console.

12.12 Tailoring Your Layout

This chapter explains how to make adjustments to your network layout through the style specific controls in the Network Layout Assistant main window. The three layout styles – Hierarchical, Circular, and Symmetric – are each covered in their own section.

12.13 Hierarchical Layout Style

- Organizes the elements in your network into a hierarchy.
- Emphasizes tree-like subnetworks in your network.
- Is excellent for maps containing buses, because the Network Layout Assistant can stretch the bus glyphs to accommodate all of the elements directly attached to them.

Figure 12-6 shows the Network Layout Assistant main window with the Hierarchical layout style default settings.

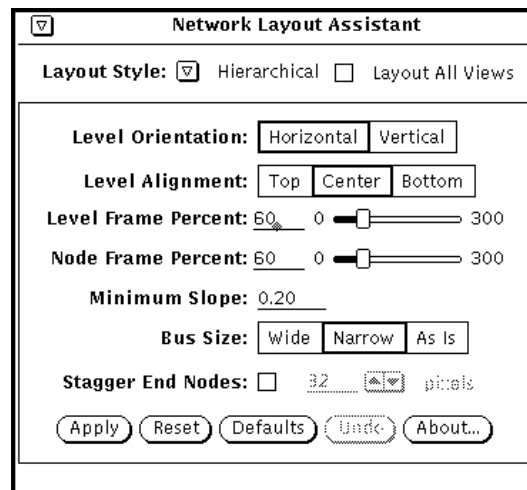


Figure 12-6 Main Window - Hierarchical Layout Style Default Settings

12.13.1 What is a Level?

A level is a grouping of nodes placed on a line. Usually a horizontal or vertical line can be drawn through the nodes that belong to a particular level. If a graph has a horizontal level orientation, each level occupies a row in the graph. If a graph has a vertical orientation, each level occupies a column in the graph.

12.13.2 Hierarchical Layout: Level Orientation

The Level Orientation panel option changes the orientation of the levels in a graph between horizontal and vertical. When the orientation is horizontal, all the nodes will line up on horizontal levels. Horizontal levels imply a top-to-bottom organization of your network in the Console. When the orientation is vertical, all the nodes will line up on vertical levels. Vertical levels cause a left-to-right organization of your network in the Console.

Figure 12-7 and Figure 12-8 illustrate the difference between horizontal and vertical orientation.

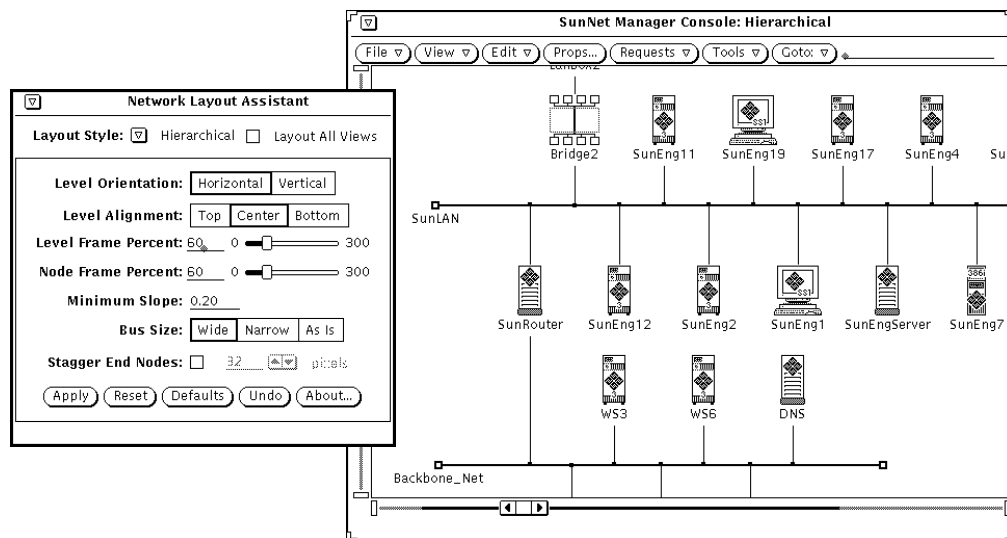


Figure 12-7 Hierarchical Layout, Horizontal Level Orientation (default)

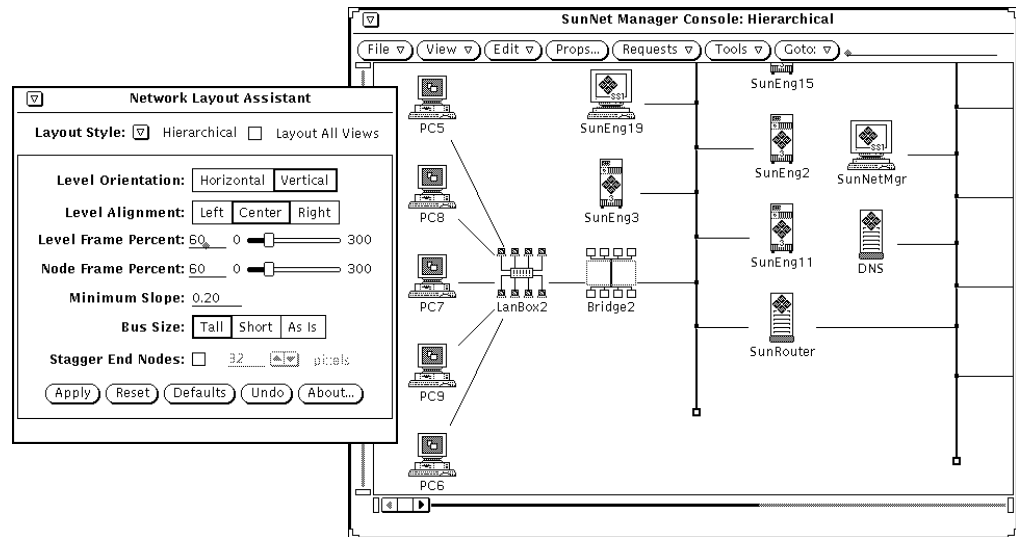


Figure 12-8 Hierarchical Layout, Vertical Level Orientation

12.13.3 Hierarchical Layout: Level Alignment

The Network Layout Assistant employs the familiar “word processing model” to specify the alignment of nodes within a level. The concept is similar to how users set the alignment of text within paragraphs in word processing software. If the levels are oriented horizontally, the Network Layout Assistant allows you to set the alignment of nodes as either top, center, or bottom-aligned within a level. If the levels are oriented vertically, the Network Layout Assistant allows you to set the alignment of nodes as either left, center, or right-aligned within a level.

This setting will only make a difference if the icons are of different sizes. With SunNet Manager 2.2 and newer versions, the supplied icons are all the same height and width and this option will have no effect.

12.13.4 Hierarchical Layout: Level Frame Percent

The Level Frame Percent control allows you to set the spacing above and below each level of glyphs in the map. This control is called “level frame percent” because the Network Layout Assistant maintains an internal “frame

level-bounding rectangle” that is a percentage larger than the smallest rectangle that can bound all of the nodes in a level. Increasing the level frame percent has the effect of spreading the levels further apart, decreasing the level frame percent has the effect of pulling the levels closer together.

Note that the glyphs in Figure 12-9 are closer together along the y axis than the glyphs in Figure 12-10 due to the smaller level frame percent.

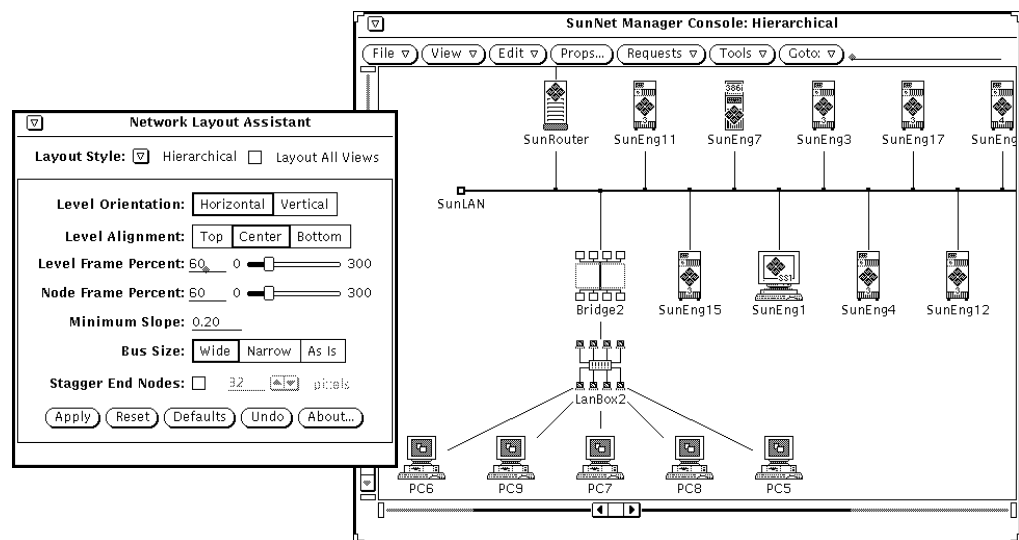


Figure 12-9 Hierarchical Layout, 60 Percent Level Frame Spacing (default)

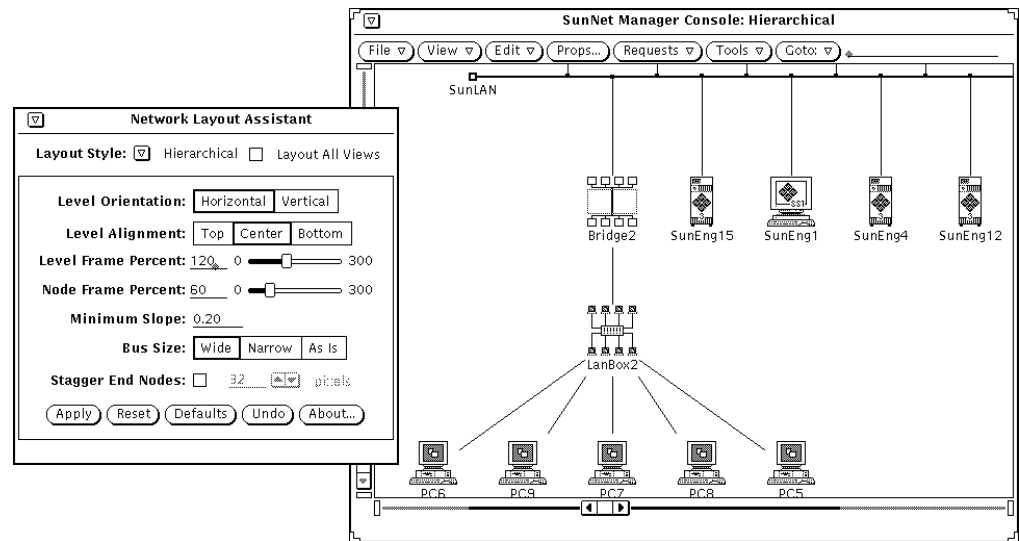


Figure 12-10 Hierarchical Layout, 120 Percent Level Frame Spacing

12.13.5 Hierarchical Layout: Node Frame Percent

The Node Frame Percent controls the spacing between nodes on the same level. This control is called “node frame percent” because the space is calculated as a percentage of the width and height of the smallest rectangle that can frame each node. Increasing this frame percentage value spreads nodes in each level further apart. Decreasing this value pulls nodes in each level closer together.

The ability to control the Node Frame Percent is useful if, for instance, the characters in your icon names overlap after a layout. If this happens, try increasing the node frame percent value.

You can see that the glyphs in Figure 12-11 are closer together along the x axis than the glyphs in Figure 12-12 due to the smaller node frame percent.

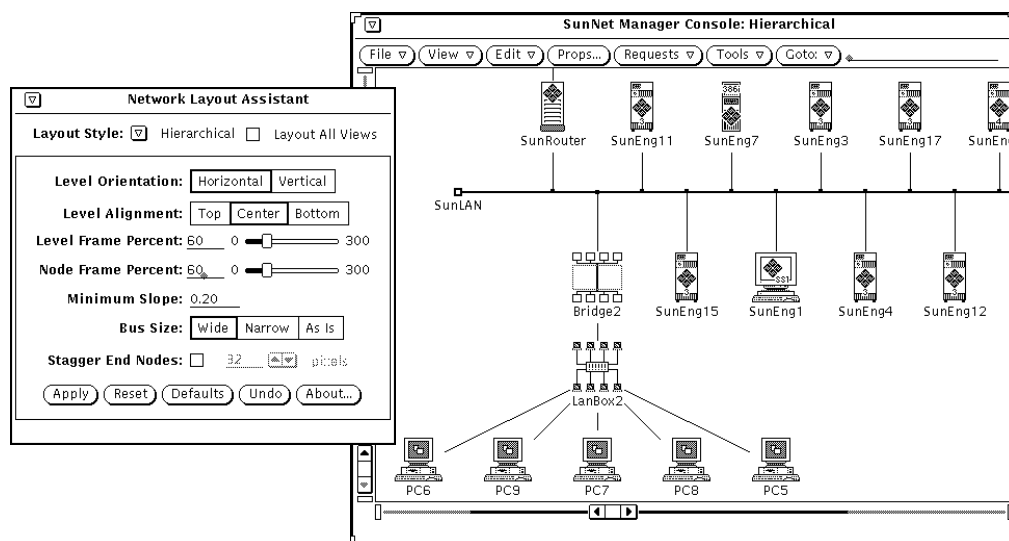


Figure 12-11 Hierarchical Layout, 60 Percent Node Frame Spacing (default)

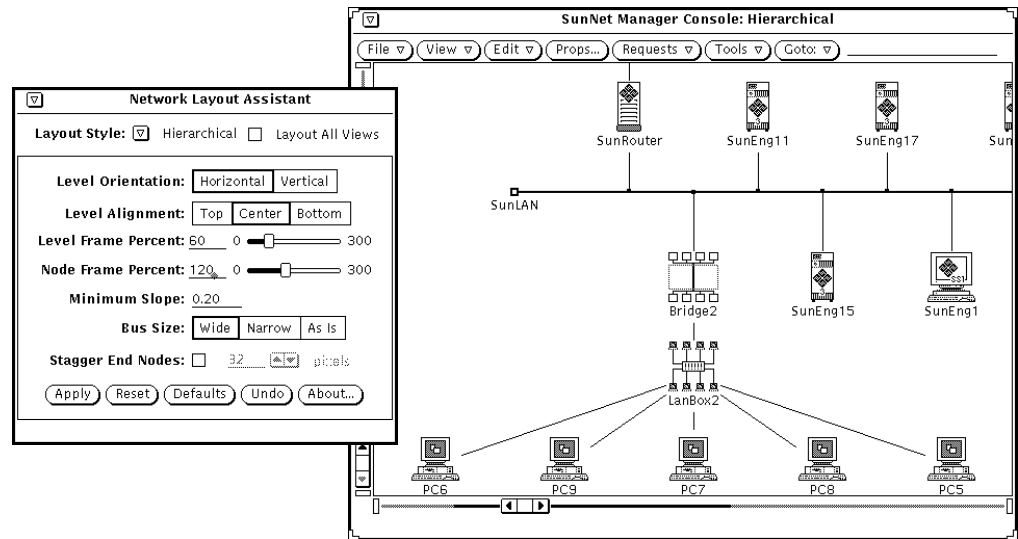


Figure 12-12 Hierarchical Layout, 120 Percent Node Frame Spacing

12.13.6 Hierarchical Layout: Minimum Slope

This control adjusts the spacing between adjacent levels to ensure that every connection has at least the minimum absolute slope. This feature is useful when networks are very dense or contain devices that have many connections.

For instance, if your network has routers or hubs with many ports, your map may have a congested area where many connections are emanating from these routers or hubs. Increasing the minimum slope can have the positive effect of increasing the distance between levels, thereby adding space between glyphs and improving the overall readability of a large network. Decreasing minimum slope, on the other hand, will allow you to show more of the map in the Console.

Figure 12-13 and Figure 12-14 demonstrates the effect of increasing the minimum slope. Notice that the glyphs in the levels above and below the *SunLAN* bus glyph are further apart in the second figure, reflecting the higher minimum slope.

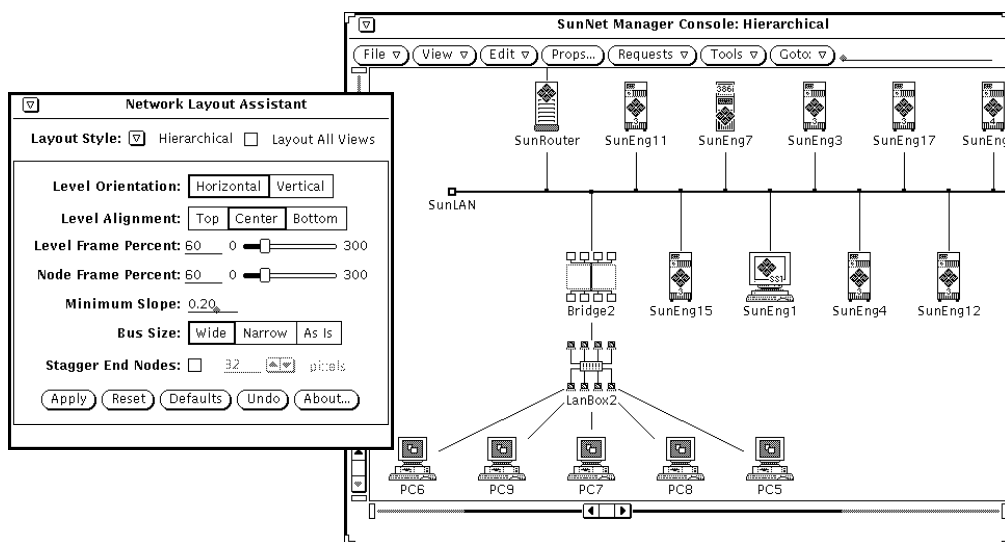


Figure 12-13 Hierarchical Layout, 0.20 Minimum Slope (default)

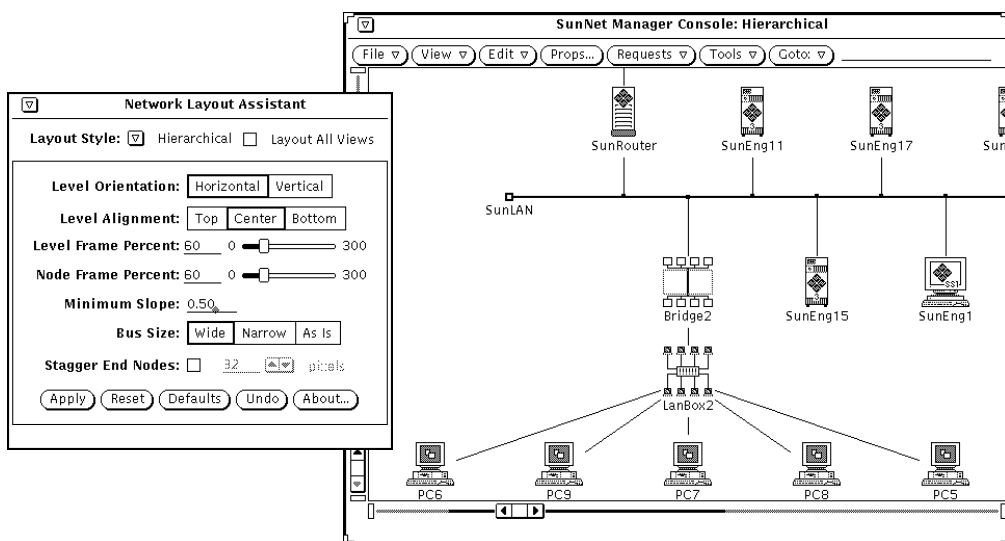


Figure 12-14 Hierarchical Layout, 0.50 Minimum Slope

12.13.7 Hierarchical Layout: Bus Size

The Bus Size option allows you to automatically resize your bus elements when you apply a layout. The choices have the following effect:

- **Wide** (Horizontal orientation), **Tall** (Vertical orientation): Your buses will be stretched to accommodate the elements directly connected to them.
- **Narrow** (Horizontal orientation), **Short** (Vertical orientation): All the buses in the view will be shrunk down to the size of an average-sized glyph.
- **As Is**: The buses will stay the same size as they were before the layout was applied.

Figure 12-15, Figure 12-16, and Figure 12-17 show how the different bus size settings affect the layout. You can see in the third figure that the “As Is” setting allows you to have different size buses in the same view.

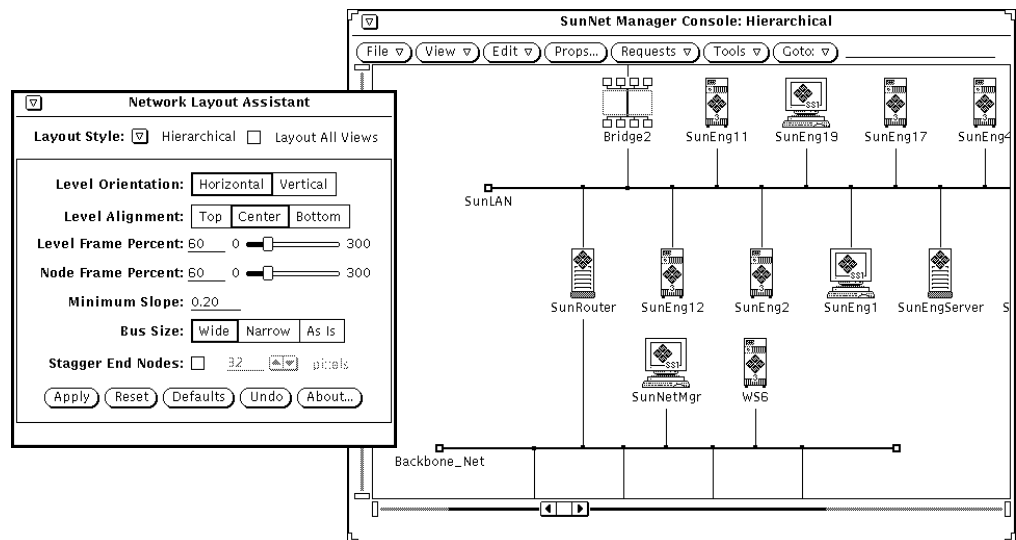


Figure 12-15 Hierarchical Layout, Wide Buses

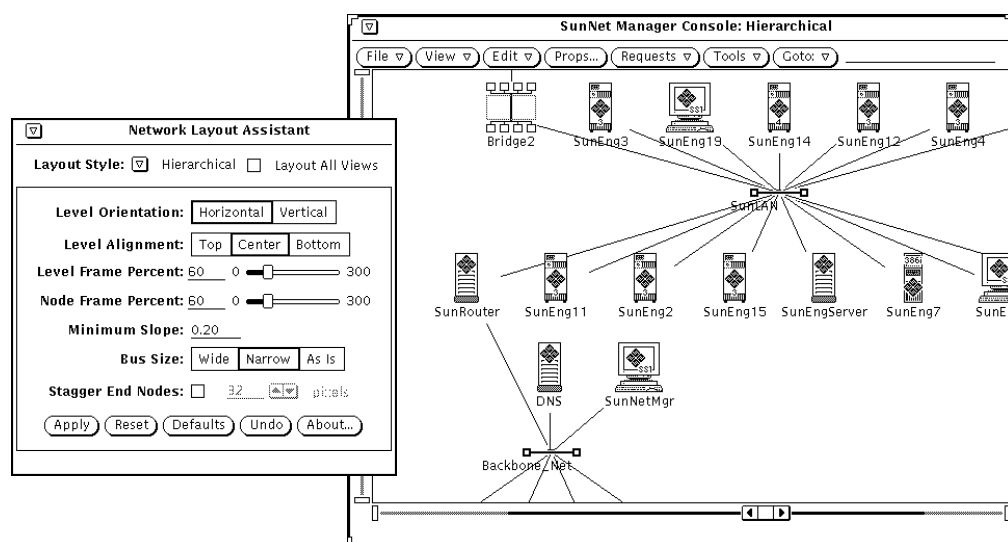


Figure 12-16 Hierarchical Layout, Narrow Buses (default)

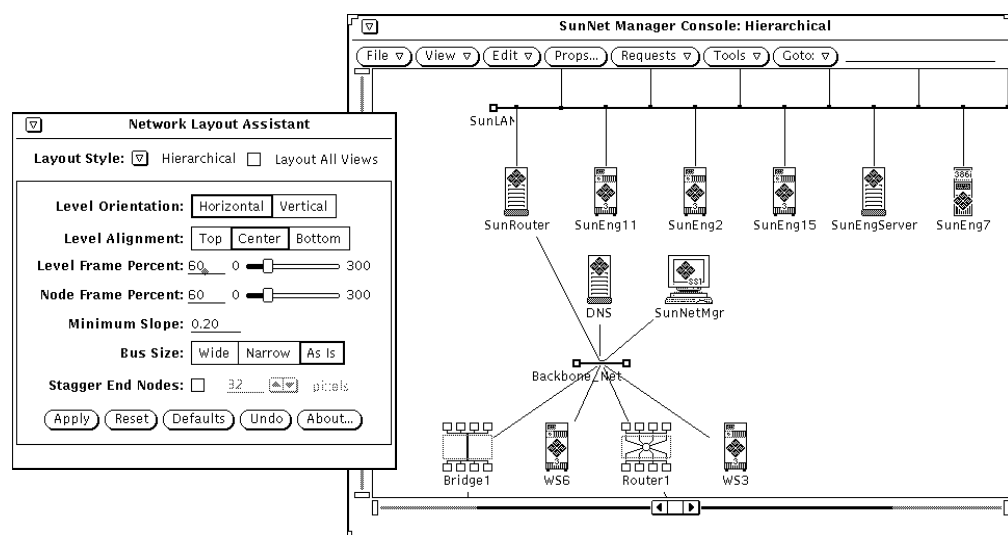


Figure 12-17 Hierarchical Layout, "As Is" Buses

12.13.8 Hierarchical Layout: Stagger End Nodes

The Stagger End Nodes control determines whether devices attached to buses are alternately staggered along a level. Specify the number of pixels to adjust the devices in the entry field provided.

12.14 Circular Layout Style

- Supports sophisticated element grouping using IP networking standards.
- Supports administrative element grouping features.
- Performs radiated placement of those groups.
- Highlights the difference between prominent backbone main sites in your network and peripheral sub-sites in your network.
- Highlights the networks within your internet by grouping nodes according to the network or subnetwork to which they belong.

Figure 12-18 shows the Network Layout Assistant main window with the Circular layout style default settings.

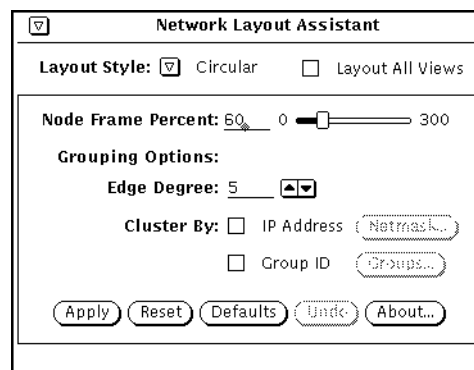


Figure 12-18 Main Window Showing Circular Layout Style Selected

12.14.1 Circular Layout: Node Frame Percent

The Node Frame Percent slider controls the spacing between each node in a cluster (circular grouping of elements). Increasing the frame percentage value spreads the elements further apart. Decreasing this value pulls the elements closer together. The ability to control the Node Frame Percent is useful if, for instance, the characters in your icon names overlap after a layout. If this happens, try increasing the node frame percent value.

In Figure 12-19 and Figure 12-20, note how the spacing between the nodes increases as the node frame percent is increased.

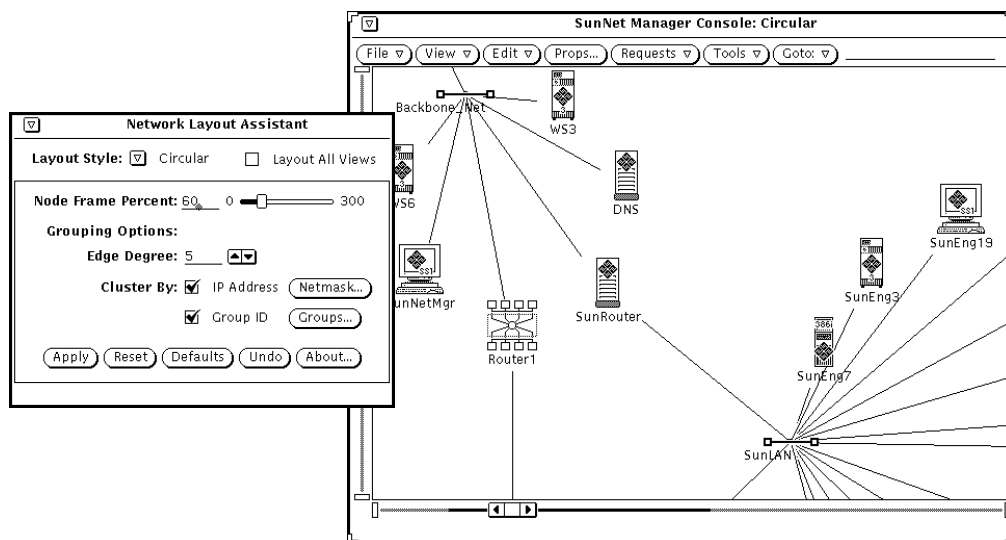


Figure 12-19 Circular Layout, 60 Percent Node Frame Spacing (Default)

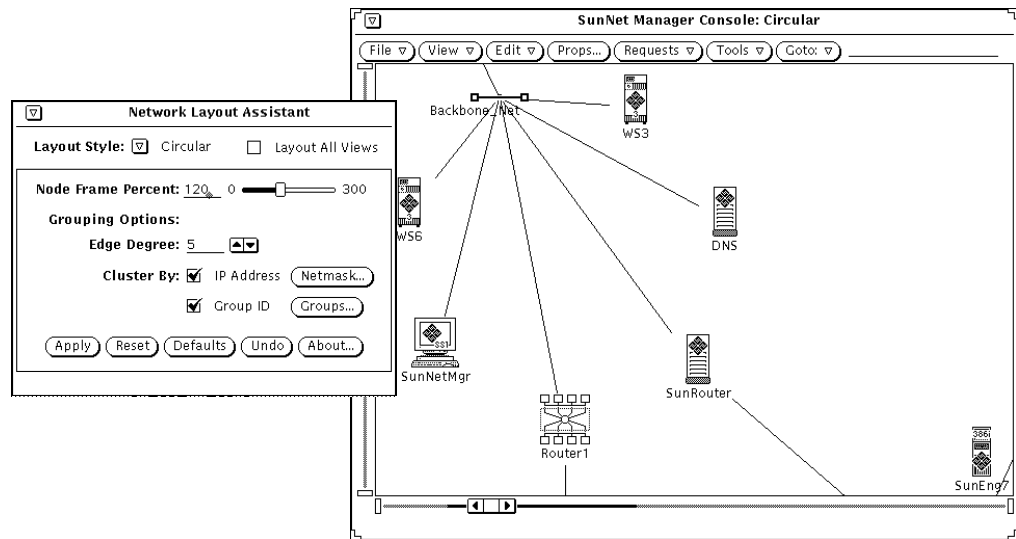


Figure 12-20 Circular Layout, 120 Percent Node Frame Spacing

12.14.2 Circular Layout: Grouping Options

The Circular layout operates by gathering elements into groups, or *clusters*, and arranging the elements in each group into a circle. The circles are then placed considering inter-group connection information.

Figure 12-21 and Figure 12-22 illustrate how Circular layout grouping can help you localize devices on a particular subnet.

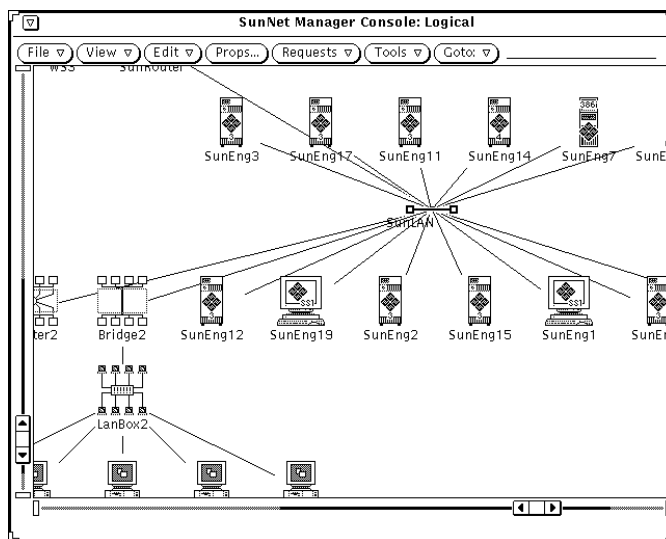


Figure 12-21 Internetwork Before Circular Layout Grouping

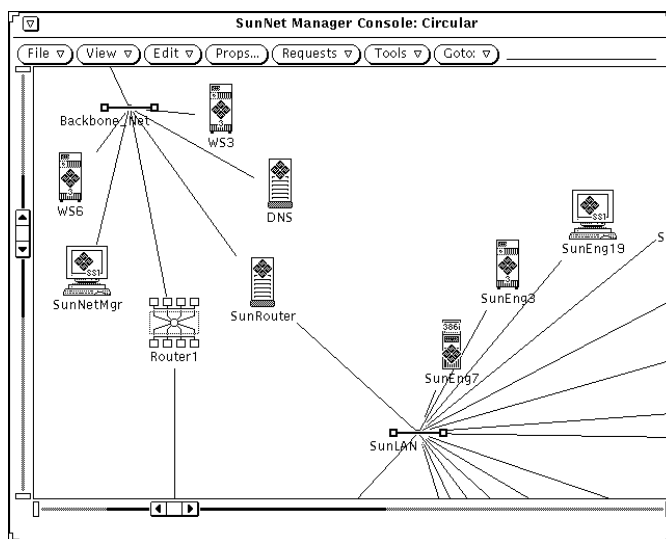


Figure 12-22 Internet After Circular Layout Grouping

The Network Layout Assistant provides three clustering mechanisms that determine which group an element is assigned to. These are:

- **Edge Degree grouping:** The default method generates groups based on topology.
- **IP Address grouping:** The IP Address of the element is used to set the group.
- **Group ID:** You explicitly set the group of a particular element.

These grouping mechanisms operate in sequence (series). If the Group ID check box is selected and a group ID exists for an element, the Network Layout Assistant organizes elements into circles based on these group ID settings. Next, if the IP Address check box is selected and an IP Address exists for an element, the element will be clustered based on the subnet mask for its network. Finally, the Edge Degree facility is used if there are any elements that have not yet been grouped by the other two methods.

The end result is that each element in the network is assigned to a group after all three grouping operations are performed.

12.14.3 Circular Layout: Edge Degree

This default clustering method uses information based on the connectivity of the network. The *edge degree* of an element is the number of connections the element has to other elements. If an element has an edge degree greater than or equal to a specified edge degree value, these elements will form the initial clusters.

For example, highly connected buses and routers should form the center of initial clusters in your network. Elements that are adjacent to the new clusters are assigned to these new clusters. Then, remaining unclustered elements with lower edge degree begin to form new clusters and attract adjacent nodes into the clusters. This operation continues until each element is assigned to a cluster.

If too many clusters appear when using the default, increase the Edge Degree number; if too few clusters appear, decrease the Edge Degree number.

The Edge Degree grouping can be overridden using the IP Address and Group ID methods described next.

12.14.4 Circular Layout: Cluster by IP Address

Most UNIX-based networks use the Internet Protocol (IP) for their network protocol. The Network Layout Assistant can make use of the elements' IP addresses when assigning groups; simply select the IP Address checkbox to cluster elements belonging to the same IP network.

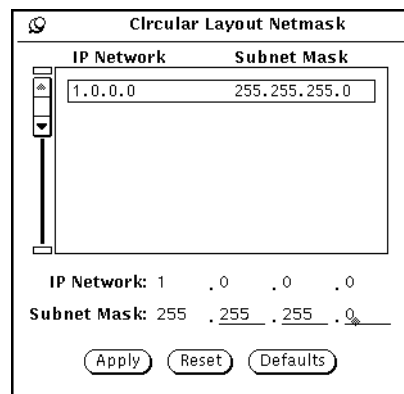
Network grouping: The Network Layout Assistant analyzes the IP address of each network element (if it has one) and gathers elements on the same Class A, Class B or Class C network into groups.

Subnet grouping: The Network Layout Assistant supports further organization of your network maps through support for IP subnetting. The Netmask window, described below, allows you to set the subnet mask for each network in your internet.

The IP Address grouping overrides the groups determined by edge degree clustering. You can further tailor the groups by using the Group ID facility described in Section 12.14.7, "Grouping Window."

12.14.5 Netmask Window

The Netmask dialog permits you to fine-tune how the elements in each network are grouped by setting a subnet mask. Subnetting is commonly used to partition IP networks into manageable groups. The netmask window allows the system administrator to apply a subnet mask value other than the default for each network. Click the “Netmask” button to pop up the netmask dialog:

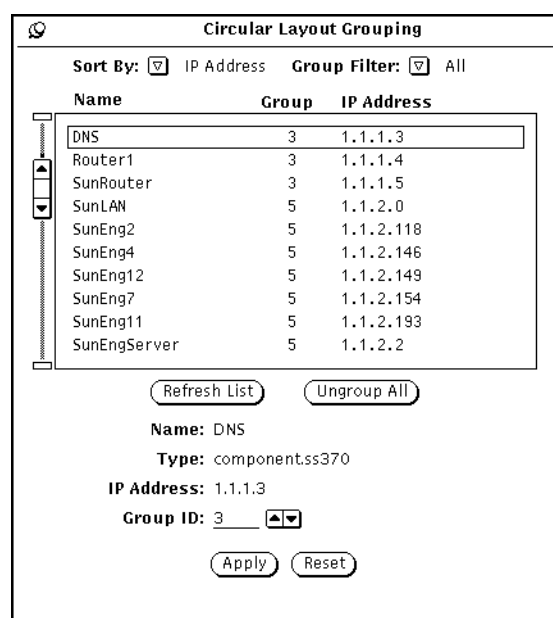


You change the mask for a network by selecting a record in the list box, entering the new mask into the entry fields and clicking Apply. Then click Apply in the Circular Layout panel to layout your internet based on the new subnet mask(s).

12.14.6 Circular Layout: Cluster by Group ID

After you apply a Circular layout using the above clustering methods, you might want to adjust the groups of a few particular elements. You can explicitly set the group of an element with the grouping window described below. If a group ID exists for an element, it will override the group set by Edge Degree clustering or IP Address clustering.

To use this grouping feature, first run a layout using Edge Degree or IP Address clustering. Then check the Group ID check box and click the “Groups” button to bring up the grouping dialog:



12.14.7 Grouping Window

The Circular Layout Grouping dialog provides fine control of element group settings. The dialog box provides a list box that displays the element name, the group to which the element belongs and the IP address of the element. You can select one or more elements in the list box and set their group.

12.14.7.1 Selecting Elements to be Grouped

You use the mouse to select items that you wish to group together (i.e. to appear on one circle). Scroll through the list box and select each element that you would like to set a group for. After selecting the items, enter a group number in the Group ID field and click the Apply button to update the group. To run the layout with the new group settings, click the Apply button on the Network Layout Assistant main window.

Note – If you use the grouping window to set an element's group ID and later uncheck the IP Address check box, the group ID(s) will be reset on the next layout.

A number of controls are provided which make using the grouping window easier:

12.14.7.2 *Sort By:*

The list box elements may be sorted by name, group or by IP address:

- **Sort By Name:** Alphabetizes the list of elements within the list box by name.
- **Sort By Group:** Sorts the list of elements by their group identifier.
- **Sort By IP Address:** Sorts the list by the IP Address of each element. Elements without IP addresses are placed at the end of the list.

12.14.7.3 *Group Filter:*

The Group Filter drop down list box provides a convenient mechanism to limit the number of items that are viewed in the list box at any time so that you can focus on the task at hand. The drop down list box provides you the following choices:

- **All:** Shows all elements that are displayed within the view.
- **Ungrouped:** Shows those elements that do not yet have groups assigned to them (group 0).
- **Group 1-N:** From this list, you select the group from which you would like to see elements listed in the list box.

12.14.7.4 *Refresh List:*

This button causes the element list box to be refreshed from the SunNet Manager database. If, for instance, you load a new database file into the Console, you will need to click the Refresh List button to see the new elements in the list.

12.14.7.5 Ungroup All:

This button causes all the elements to be reset to group 0 (ungrouped).

12.15 Symmetric Layout Style

This style is:

- Ideal for organizing wide area meshed networks.
- Highlights the symmetries inherent in the topology of the network.
- Lays out any network regardless of routing protocol(s) employed or discovery mechanism.
- Provides uniform distribution of glyphs on the display.

Figure 12-23 shows the Network Layout Assistant main window with the Symmetric layout style default settings.

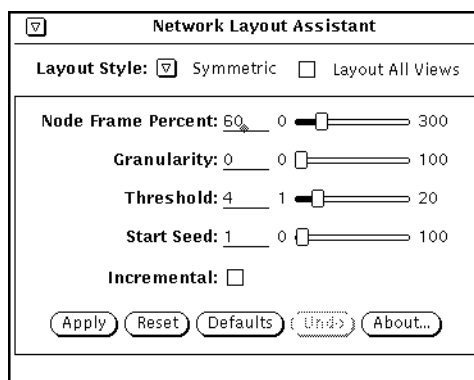


Figure 12-23 Main Window Showing Symmetric Layout Style Selected

12.15.1 Symmetric Layout: Node Frame Percent

The Node Frame Percent controls the spacing around each glyph. A higher node frame percent value increases the cushion of space around the elements and lengthens the connections between glyphs. Smaller values result in more condensed layouts with shorter connections.

Notice how the glyphs in Figure 12-24 are closer together than the glyphs in Figure 12-25 due to the smaller node frame percent.

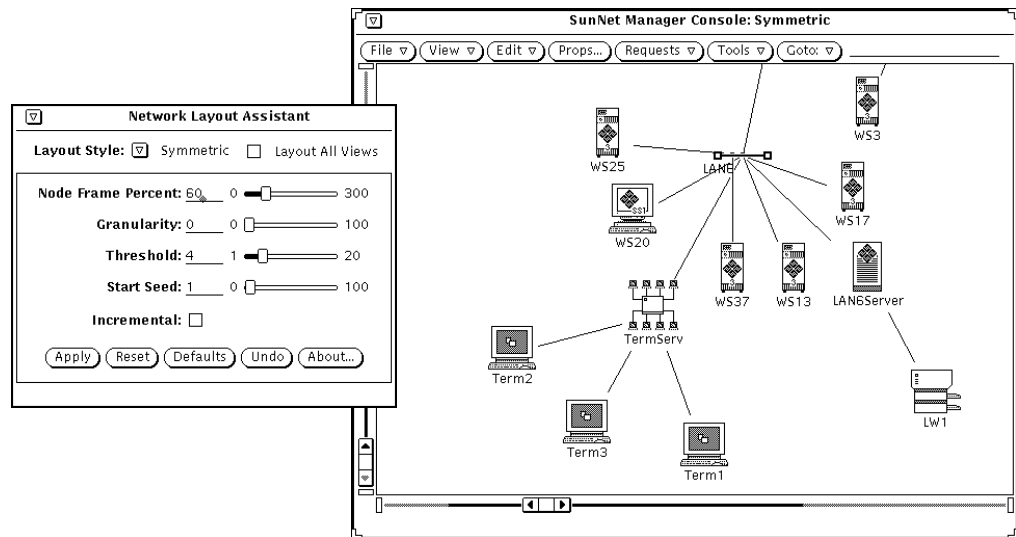


Figure 12-24 Symmetric Layout, 60 Percent Node Frame Spacing (Default)

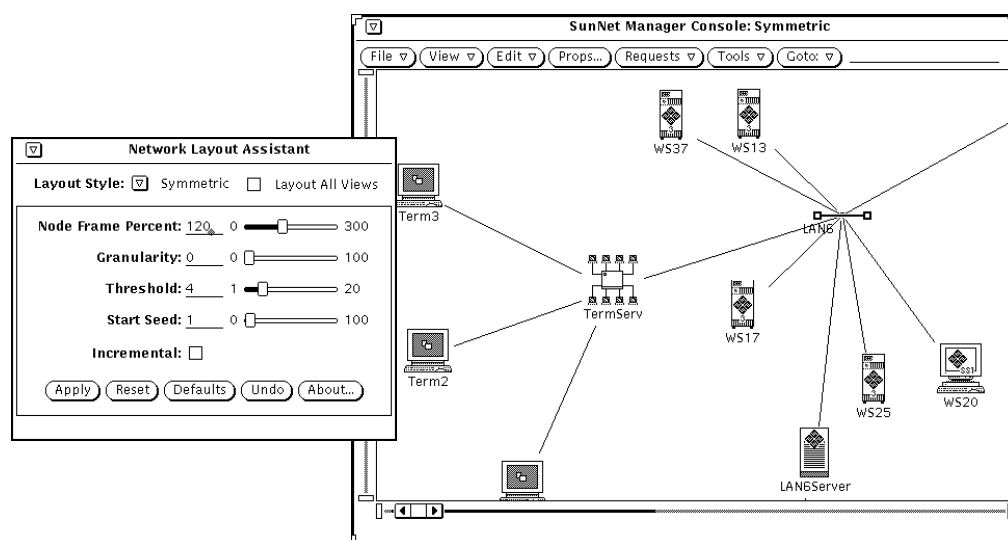


Figure 12-25 Symmetric Layout, 120 Percent Node Frame Spacing

12.15.2 Symmetric Layout: Granularity

The Granularity option controls the size of the underlying coordinate grid upon which the network is laid out. At the default of 0, the Symmetric style uses a relatively fine grid. A value of 100 specifies a very coarse grid. The difference in granularity is reflected in a trade-off between the final layout quality and computation time. A fine grid generally produces a better layout but takes longer to compute. A coarse grid generally produces a rougher layout, but runs much faster.

If speed is a consideration, set a higher granularity to produce draft layouts, and then reduce the granularity when you want to produce a presentation-quality layout.

12.15.3 Symmetric Layout: Threshold

The threshold determines the point at which Symmetric layout processing stops and outputs a final layout. Lower threshold values generally produce better layouts but usually take more time to compute.

The default threshold value of four is adequate for most networks. Lowering the value to two or one can produce better results while taking longer to compute. Increasing the value will produce slightly lower quality layouts but will be faster to compute.

12.15.4 Symmetric Layout: Start Seed

The Start Seed affects the starting configuration of the map before the layout process begins. Each different seed value produces a unique final layout. While the result of changing the start seed is admittedly random, this control can help you generate an better layout.

Figure 12-26 and Figure 12-27 show the same topology with different start seeds.

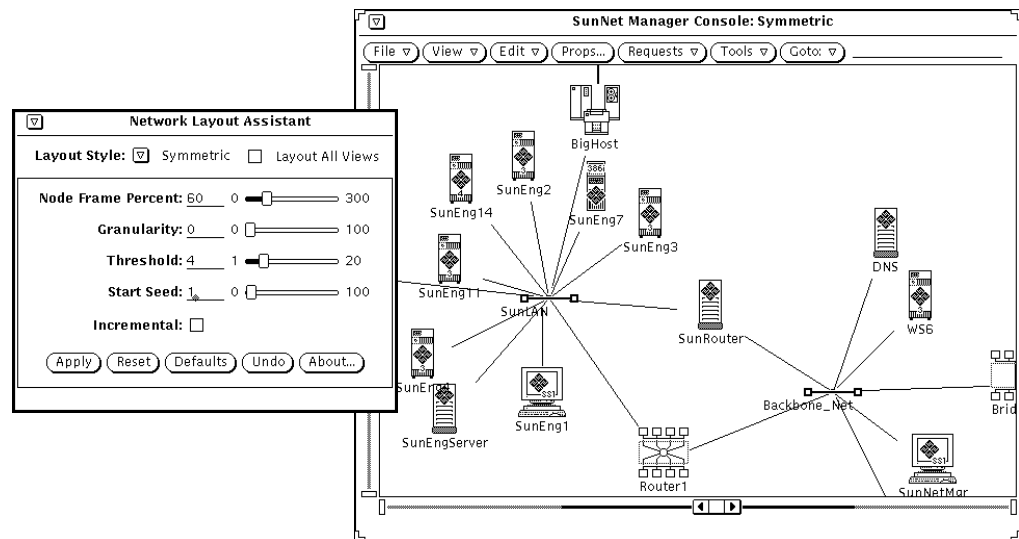


Figure 12-26 Symmetric Layout, Start Seed of 1 (default)

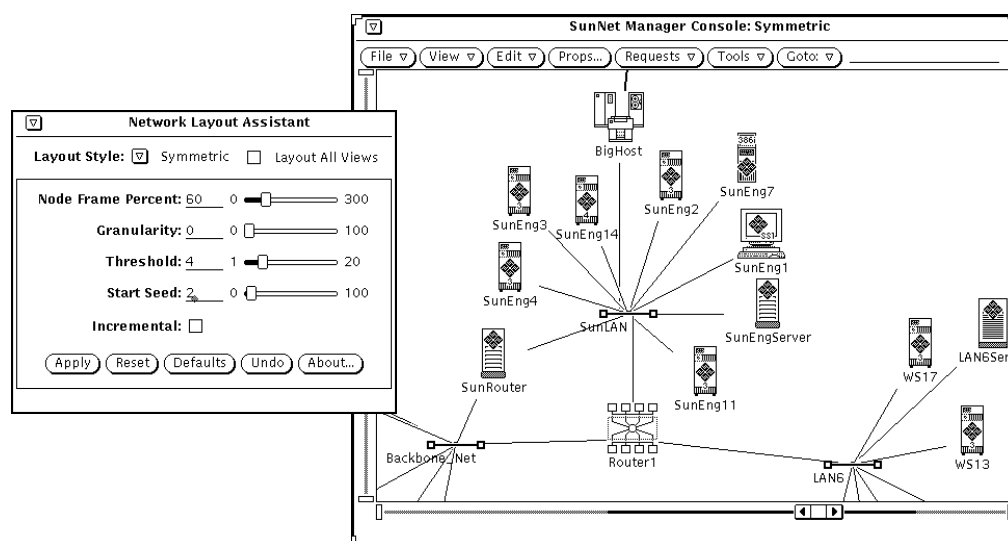


Figure 12-27 Symmetric Layout, Start Seed of 2

12.15.5 Symmetric Layout: Incremental

If this check box is selected, the Symmetric layout style will use the current positions of the nodes as a starting point when determining the new layout. The incremental setting helps you maintain correlation between layouts, and will sometimes improve the layout quality if you run multiple layouts.

12.16 Network Layout Assistant Restrictions

If you start multiple SunNet Manager Consoles on the same desktop, you need to set the “Display user name in Title” checkbox in the Console’s “Props...” window.

You cannot create view names that contain a colon followed by a space (“:”).

The Program name used to start the SunNet Manager Console must be named `snm`.

Part 2 — Reference

13.1 Overview

This Reference Part describes the features and functions of SunNet Manager in more detail, including the following:

- Agents and proxy agents
- Ancillary daemons (activity daemon and event dispatcher)
- The details of sending data and event requests
- The details of viewing data and event requests
- Console properties
- The management database
- SNMP support
- Results Browser
- Results Grapher
- IP Discover
- IPX Discover
- Set Tool

13.2 Agents and Proxies

There are two types of SunNet Manager agents: those that directly access managed objects and those that indirectly access managed objects. Most of the agents provided with this release manage objects on the Sun workstations where they are installed. For example, the `hostmem` agent uses the same mechanism as `netstat -m` to get memory utilization data.

The second type of agent provides the ability to manage objects that reside in other Sun workstations or in other vendors' devices. Such agents are called *proxy agents*. Proxy agents run on Sun workstations, called *proxy systems*, and use protocol translation mechanisms to provide the necessary access to the managed objects. The proxy system may be the workstation on which the SunNet Manager Console is running or another workstation on the network. The proxy system can also be a workstation in a different subnet or domain from where the Console is running. This allows SunNet Manager to extend into virtually any domain.

13.2.1 Agents Included with SunNet Manager

A Console can manage both SunOS 5.x, and SunOS 4.x clients. All SNM agents that run on SunOS 4.x clients are also available for the Solaris 2.x environment. Note however, that certain of the agents that run on Solaris 2.x clients support different attributes than their SunOS 4.x counterparts. The Solaris 2.x versions of these agents have been renamed to `<agentname>2`. For example, `na.iostat2` is the Solaris 2.x version of the (SunOS 4.x) agent `na.iostat`. See Chapter 2, "Planning for Network Management," for a list of agents and proxy agents shipped with the current release.

Note - `snmp`, `snmp-mibII`, and `sun-snmp` and several RFC schema files can be used with the SNMP proxy agent `na.snmp`. For each of the other agents shipped with the current product, there is only one corresponding schema file.

Note - `na.hostif` is the same for SunOS 4.x and Solaris 2.x except that the metric table is inaccessible for Solaris 2.x.

Table 13-1 lists the agents that have substantial differences between their SunOS 4.x and Solaris 2.x versions. Table 13-1 shows the attributes that were added to the Solaris 2.x agents; it also shows attributes from the SunOS 4.x version that were deleted for the Solaris 2.x version of these agents.

Table 13-1 Agents Specific to Solaris 2.x

Agent	SunOS 4.x attributes removed	New attributes added for Solaris 2.x
na.etherif2	input group: ibufs ibufdesc idiscard iframedesc output group: ojams odrops watchdog obufdesc copies nocfs obufs oframedesc obuferr	input group: imissed inocanput output group: oinits
na.hostmem2	mbuf group mbuf_uses group streams group: mblkused mblkfree mblk% mblkcom mblkfail dblks table	streams group: msgused msgfree msg% msgcum msgfail linkblkused linkblkfree linkblk% linkblkcum linkblkfail streveused strevefree streve% strevecum strevefail

Table 13-1 Agents Specific to Solaris 2.x

Agent	SunOS 4.x attributes removed	New attributes added for Solaris 2.x
na.iostat2	disk table: mbytes xfers seeks activeT xferT seekT %xferT avgxferT kpbs	disk table: kreads kwrites svcwait acttrans svctime %wait %busy
na.layers2	no changes	no changes
na.cupstat		available only on Solarix 2.x

These agents represent examples of the data you can gather using SunNet Manager, and are not meant to be an exhaustive set. To write your own agent, refer to your *Site/SunNet/Domain Manager Application and Agent Development Guide*.

Many Sun products include agents that can be used with SunNet Manager. Refer to your product documentation or call your local Sun representative for more information.

13.2.2 Ancillary Daemons

SunNet Manager includes two ancillary daemons: the activity daemon (`na.activity`) and the event dispatcher (`na.event`). The activity daemon is a process that uses the Manager/Agent Services to ensure ongoing requests (activities) continue to be serviced. The activity daemon uses the activity log, a record of active requests started from the Console.

The event dispatcher is a process that uses the Manager/Agent Services to direct reports of events to the proper destinations (*rendezvous process*), such as the SunNet Manager Console. Any other management application can use the event dispatcher, registering to receive some or all event reports based on a variety of selection criteria. The Event Dispatcher also logs all event reports in the Event/Trap Log. The Event/Trap Log is in ASCII format, so you can examine and modify it using standard UNIX tools.

Four processes of the event dispatcher are started when you invoke the Console. When you make the first request to any agent, two processes of the activity daemon are started. When you exit the Console, two processes of `na.activity` and one copy of `na.event` remain.

13.3 SunNet Manager Directories and Files

This section discusses the files and directories that are created during SunNet Manager installation. The Solaris 1.x and Solaris 2.x versions of SunNet Manager differ in the default locations assigned to these files and directories by the installation scripts. The default locations for these files are summarized in Table 13-2.

The `snm.conf` file is the configuration file used by SNM agents and daemons. (For Solaris 1.x installations of SNM, `snm.conf` is located in `/etc`; for Solaris 2.x installations `snm.conf` is located in `/etc/opt/SUNWconn/snm`.) See the `snm.conf(5)` man page for more information about changing keyword values.

The SunNet Manager installation script prompts you for the following locations:

- The directory where the SNM software will be installed. This normally is `/usr/snm` for Solaris 1.x installations; for Solaris 2.x installations the default location is `/opt/SUNWconn/snm`.

If you install SNM into a directory other than the default, you must set the environment variable `SNMHOME` to the directory where SNM is installed before you can use the software.

- The directory where the SNM database will be written. (If you do not specify otherwise, this is `/var/adm/snm` for Solaris 1.x installations; the default location for Solaris 2.x installations is `/var/opt/SUNWconn/snm`.)

Caution – You should have at least 10 Mbytes of space for the initial database directory. You can change the location of the database directory after installation by setting the environment variable `SNMDBDIR` to a writable directory.

- The directory where SNM log files will be written. (If you do not specify otherwise, this is `/var/adm/snm` for Solaris 1.x installations; the default location for Solaris 2.x installations is `/var/opt/SUNWconn/snm`.)

The log files grow during Console operation. This could cause a disk space problem if the `/var` directory is part of the root file system. You can change the location of the log files after installation by specifying fully-qualified path names for the keywords `activity-log`, `event-log`, `monitor-log`, and `request-log` in the `snm.conf` file. Make sure that the new directories are writable by root or mode `777` so that the Console can write to the log files.

13.3.1 Installation Directory Files

If you install the SNM software in the default path (`/usr/snm` for Solaris 1.x installations, `/opt/SUNWconn/snm` for Solaris 2.x installations), the installation script will create the following directories and files under the default directory:

- `agents` directory, which contains SNM agents and their corresponding schema files. You can add your own agent and schemas to this directory. This directory also contains the files `enterprises.oid`, which contains mappings for enterprise identifiers, and `snmp.oid`, which contains mappings for SNMP MIB I object identifiers.
- `bin` directory, which contains binaries used by SNM. One of the binaries, `snm_asroot`, is undocumented.
- `icons` directory, which contains SNM icon files. You can add your own icons to this directory.
- `include` directory, which contains header files for building agents.
- `lib` directory, which contains SNM libraries.
- `man` directory, which contains SNM man pages.
- `5.x`, which contains the Solaris 2.x version of the SunNet Manager agents, for managing SunOS 5.x machines.
- `src`, which contains sample source code for selected agents and manager applications.
- `struct` directory, which contains the following files:
 - `elements.schema` defines the element types. If you need to add site-specific elements, create your own file with the `.schema` extension (for example, `myelements.schema`).

- `netware_elements.schema` contains Novell specific element definitions.
- `snm.schema` contains NLA specific information.
- `example.db` is an example database that you can load into the Console.
- `snm.glue` is a file that contains a starting set of definitions for the Console. Do *not* modify this file.

13.3.2 Database Files

A database directory is created in the path specified by the environment variable `SNMDBDIR`. If `SNMDBDIR` is not specified, then the database path specified during installation is used. The database directory is `db.<user_name>`, where `<user_name>` is the value of the environment variable `SNM_NAME`, or the environment variables `LOGNAME` or `USER`, if `SNM_NAME` is not set. The following database files are created for SNM:

- `snm+lock` is the Console lock file; this prevents more than one user from accessing the same runtime database.
- `snmdb+lock` is the database API lock file; this file enables the database to be locked when entries are changed programmatically.
- The files `events.ind`, `events.rec`, `nc.ind`, and `nc.rec` make up the runtime database used by the Console.
- The following data files, which are used by IPX Discover, reside in your database directory: `dumpfile`, `snm_index.dat`, `x.x.x.x.dat`, `nc.ind`, and `nc.rec`, where `x.x.x.x.dat` is an `ipaddress.dat` file. An example would be `129.9.119.5.dat`

13.3.3 Log Files

Log files are created in the directory specified during installation. You can also change the location of individual log files after installation by modifying the appropriate log file entry in the `snm.conf` file. The following log files are created for SNM:

- `activity.log` is a log of active requests started from the SNM Console host; this file is used by the activity daemon.
- `event.log` is the log file written by event dispatcher.

- `monitor.log` is written by the `na.logger` agent. This agent logs data reports launched by the `snm_cmd` command.
- `request.log` is a record of requests that are to be restarted by the appropriate agent; this file is used by the agents.

13.3.4 Miscellaneous Files

- The `snmp.hosts` file contains information about SNMP hosts. (The default location of the `snmp.hosts` file for Solaris 1.x installations is `/var/adm/snm`; for Solaris 2.x installations the default location is `/var/opt/SUNWconn/snm`.) You can add entries into this file to specify enterprise-specific host information.
- The `snmp.traps` file contains information about enterprise-specific traps. (For Solaris 1.x installations the default location is `/var/adm/snm`; for Solaris 2.x installations, the default directory is `/var/opt/SUNWconn/snm`.) You can add entries into this file to specify enterprise-specific trap information.
- The `$.HOME/.SNMdefaults` file contains information pertaining to SNM Console properties. For example, when the icon size is set to a new value (i.e., 32X32), or the schema path changed, these values and other properties are written to your `$.HOME/.SNMdefaults` file. If you subsequently remove SNM, then reinstall it (e.g., migrating from a previous version to the current version), the old `.SNMdefaults` values will continue to be used. You can modify the values by using the SNM Console Props button, or by editing your `.SNMdefaults` file. You can ensure that the Console property defaults are used by moving a preexisting `.SNMdefaults` file from your home directory or deleting your `$.HOME/.SNMdefaults` file.
- The `.SNMpredefined` file contains predefined data and event request records provided with the current product. (The default location of this file for Solaris 1.x installations is `/usr/snm/struct`; for Solaris 2.x installations, the default path is `/opt/SUNWconn/snm/struct`.) For information about creating and sending predefined data and event requests, refer to Chapter 18, “Management Database.” When you modify or create predefined data and event requests, they are stored in your `$.HOME/.SNMpre-defined` file along with the predefined requests provided with this product.

- The `linkmap` file is an ASCII file that contains information used by the SNM Console to provide link management capabilities. (The default location of the `linkmap` file is `/var/adm/snm` for Solaris 1.x installations and `/var/opt/SUNWconn/snm` for Solaris 2.x installations.) It is either created by the network manager or the IP Discover tool; the IP Discover tool will populate this file with discovered links (if 2 or more hops are specified as the scope of a network discovery operation). Refer to the `linkmap (5)` manual page for information about the format and use of the `linkmap` file.
- The `.snmautomanagement` file contains information about custom auto requests.

13.4 Environment Variables Used with SunNet Manager

You can use the following environment variables with SunNet Manager:

- The `PATH` environment variable should include `<installation_path>/snm/bin` where:
 - `<installation_path>` will be `/usr` if SNM is installed in the default location for SunOS 4.x installations.
 - `<installation_path>` will be `/opt/SUNWconn` if SNM is installed in the default location for the Solaris 2.x version.
- If the software for the current product is installed in a directory *other* than the default location (`/usr/snm` for Solaris 1.x, `/opt/SUNWconn/snm` for Solaris 2.x), you *must* set the `SNMHOME` environment variable to the installation directory.
- The `HELPPATH` environment variable should include `<installation_path>/snm/help` where:
 - `<installation_path>` will be `/usr` if the default location is used for a Solaris 1.x installation.
 - `<installation_path>` will be `/opt/SUNWconn` if the default location is used for a Solaris 2.x installation.

This enables you to get SunNet Manager on-line help.

- The `MANPATH` environment variable should include `<installation_path>/snm/man` where:
 - `<installation_path>` will be `/usr` if the default location is used for a Solaris 1.x installation.

- *<installation_path>* will be `/opt/SUNWconn` if the default location is used for a Solaris 2.x installation.

This enables use of the SunNet Manager man pages.

- The `SNM_NAME` environment variable can be used to allow multiple instances of the Console and database to be run from a single user ID. One or more names can be specified with `SNM_NAME`. Each instance of the Console is associated with its own runtime database, located in the `db.<snm_name>` directory, where *<snm_name>* is a name specified by the environment variable `SNM_NAME`. Each instance of the Console will use the user's `$HOME/.SNMdefaults` and `$HOME/.SNMpre-defined` `.snmanagement` files.
- The `SNMDBDIR` environment variable is used to specify the directory where the predefined data and event request ASCII data record file resides. `/var/adm/snm` is the default directory for the Solaris 1.x version of the current product; the default location for Solaris 2.x is `/var/opt/SUNWconn/snm`.
- The `SNMLINKMAP` environment variable is used to specify the directory where the `linkmap` file used by the link management feature is located. If this environment variable is undefined, the default location is:
 - `/var/adm/snm` for the Solaris 1.x version
 - `/var/opt/SUNWconn/snm` for the Solaris 2.x version.
- The `SNMDISCOVERMAP` environment variable is used to specify the directory where the `discover.conf` file, used by the IP Discover Tool, is located. `/var/adm/snm` is the default directory for Solaris 1.x installations; `/var/opt/SUNWconn/snm` is the default location for Solaris 2.x installations.

Table 13-2 Summary of Default SNM File Locations

Files	Default Location in Solaris 1.x Version	Default Location in Solaris 2.x Version
Agents and schema files	<code>/usr/snm/agents</code>	<code>/opt/SUNWconn/snm/agents</code>
Agents for Solaris 2.x machines	<code>/usr/snm/5.x</code>	<code>/opt/SUNWconn/snm/5.x</code>
<code>discover.conf</code> file	<code>/var/adm/snm</code>	<code>/var/opt/SUNWconn/snm</code>
<code>elements.schema</code> file	<code>/usr/snm/struct</code>	<code>/opt/SUNWconn/snm/struct</code>

Table 13-2 Summary of Default SNMP File Locations

Files	Default Location in Solaris 1.x Version	Default Location in Solaris 2.x Version
Example database	/usr/snm/struct	/opt/SUNWconn/snm/struct
Header files	/usr/snm/include	/opt/SUNWconn/include
Icon files	/usr/snm/icons	/opt/SUNWconn/snm/icons
linkmap file	/var/adm/snm	/var/opt/SUNWconn/snm
Log files	/var/adm/snm /var/adm/snm	/var/opt/SUNWconn/snm /var/opt/SUNWconn/snm
Manager Services libraries	/usr/snm/lib	/opt/SUNWconn/snm/lib
man pages	/usr/snm/man	/opt/SUNWconn/snm/man
messages file	/var/adm/snm	/var/opt/SUNWconn/snm
Predefined data/ event request file	/var/adm/snm	/var/opt/SUNWconn/snm
On-line help files	/usr/snm/help	/opt/SUNWconn/snm/help
Sample agent and application source code files	/usr/snm/src	/opt/SUNWconn/src
snm.conf file and snmpd.conf file	/etc	/etc/opt/SUNWconn/snm
.SNMdefaults file	\$HOME	\$HOME
SNM executables	/usr/snm/bin	/opt/SUNWconn/snm/bin
snm.glue file	/usr/snm/struct	/opt/SUNWconn/snm/struct
.SNMpredefined file	/usr/snm/struct	/opt/SUNWconn/snm/struct
snmp.hosts file	/var/adm/snm	/var/opt/SUNWconn/snm
snmp.traps file	/var/adm/snm	/var/opt/SUNWconn/snm
.snmautomangement	\$HOME	\$HOME

13.5 *Extending SunNet Manager*

The SunNet Manager infrastructure provides a platform extensible to any application. The SunNet Manager Console user interface is an application that uses the underlying services. SunNet Manager also includes a command-line mechanism for initiating data and event reporting called `snm_cmd(1)`.

You will probably want to extend the capabilities of SunNet Manager. Some of the mechanisms to support this are discussed later in this document. Refer to your *Site/SunNet/Domain Manager Application and Agent Development Guide* to implement the extensions you need.

This section discusses the following topics:

- Freezing the Console
- Control panel buttons and menus
- Element Glyph menu
- Data request Properties window
- Event request Properties window
- Viewing and modifying request results

14.1 SunNet Manager Console

The SunNet Manager Console is the central management application in the SunNet Manager package—the place where you initiate management tasks and management information is returned.

14.1.1 Management Database

The Console relies on the definitions and information contained in a management database (MDB). Emerging open management standards (for example, OSI and SNMP) specify that agents abstract the properties (or attributes) of managed objects into data items (for example, “how busy a CPU is” may become a value between 0 and 100). In SunNet Manager, the attributes

of a managed object are described in a portion of the MDB called the agent schema. The agent is able to respond to the manager's request because both use the same data definitions for the managed object.

Each *instance* of the Console requires a *separate*, runtime database. In other words, multiple Console instances *cannot* share a single runtime database.

The MDB data and any updates made using the graphical editing capabilities constitute a dynamic runtime database, which may be saved to an ASCII database file at any time. The ASCII format supports easy modification and portability to other systems. See Chapter 18, "Management Database," for more on MDB files and how to modify them.

SNM provides an application programming interface (API) that allows an application program to query or modify the runtime database. The API functions are described in your *Site/SunNet/Domain Manager Application and Agent Development Guide*.

14.1.2 Graphical Interface

The SunNet Manager Console presents an object-oriented interface which may be tailored to depict a particular management domain. The Console uses the OPEN LOOK Graphical User Interface running under OpenWindows™ 3.0 or later, Motif, or Window Manager. OpenWindows supports the X11 protocol, which allows the Console and other application windows to be displayed on a network display that is managed by an X11 server. X terminals must be fully compatible with the MIT X11 release 4 server and use an OPEN LOOK-compliant window manager.

The SNM Console supports X terminals by allowing multiple instances of the Console, using separate runtime databases, to run on a single machine at the same time. Each instance of the Console is mapped to the name of the user who invokes it. Each instance of the Console will only work with other SNM tools that use the same user name.

14.1.3 Data and Event Reporting

The Console provides mechanisms for initiating requests for *data* reporting and *event* reporting. Data reporting allows you to direct agents to send reports of raw management data on a periodic basis. Event reporting allows you to direct agents to report only when specified conditions are met (that is, when an event occurs).

You direct an agent by making a request in which you specify the parameters for a desired management task. The request contains information on the object to be managed and how often the agent is to report.

When requesting data reports, you can elect to gather data in groups or individually, have data reported immediately or on a deferred basis, and store returned data in files. For event reports, you have many options for setting the condition(s) of the event.

The Console also supports the display of reported data and event indications (including audible, visual, and programmatic mechanisms). Visual changes resulting from predefined event conditions are propagated through the Console so that you can see at a glance if an event has occurred. From the Console, you can also use the Set Tool, a window interface that allows you to change attribute values. Currently, only the Simple Network Management Protocol (SNMP) proxy agent supports Set Tool operations. With the Set Tool, you can request to change (set) the value of one or more attributes in a group or among different groups. The Console can also be extended to support user specified commands as well.

Context sensitive help is available on-line for Console screens through popup Help windows.

14.1.4 Console Tools

SunNet Manager includes several tools that you can invoke from the Console:

- The IP Discover tool seeks out network elements and automatically creates a graphical representation of your network. As it finds elements, Discover adds them to the runtime database. Refer to Chapter 22, “IP Discover,” for a description of the Discover functions.

- The IPX Discover tool seeks out IPX network elements and topology, discovers services these nodes provide, and automatically creates a graphical representation of the network. As it finds elements, IPX Discover adds them to the runtime database.
- The Results Browser allows you to examine and organize log files. See Chapter 20, “Browser,” for Browser functions.
- The Results Grapher allows you to visualize data reports and log file information. You can send data directly from data reports to the Grapher, or you can send log file data from the Browser to the Grapher. See Chapter 21, “Results Grapher,” for Grapher functions.
- The Set Tool allows you to change SNMP attribute values. Refer to Chapter 24, “Set Tool,” for more information about Set Tool.

Context-sensitive help is available on-line for these tools through popup Help windows. See your installation guide for information about the environment variable required for SNM on-line help.

14.2 Freezing the Console (Read-Only Mode)

Starting with version 2.3, you can place the console in read-only mode so that object additions, deletions, and updates will not be allowed through the console or through any management application.

Consistent with previous versions of SunNet Manager, read-only state allows only one instance of the console to be active for a given runtime database.

In read-only mode, you can run your applications and manage requests. However, the following menu items are dimmed: Edit, Create Predefined Requests, and Change Type.

To enable read-only mode:

1. **Open Properties in the Console menu.**
2. **Click SELECT on Miscellaneous from the Category pulldown item.**
3. **Click SELECT on the read-only mode button.**
4. **Click SELECT on Apply.**

14.2.1 Topology

When the Console is in Read-Only mode, see Figure 14-1, you cannot move or delete topology objects. No new objects can be added or updated, nor will you be able to create new predefined requests. The type of object cannot be changed, and appropriate menu items are dimmed. You can still send data and event requests and configure devices.

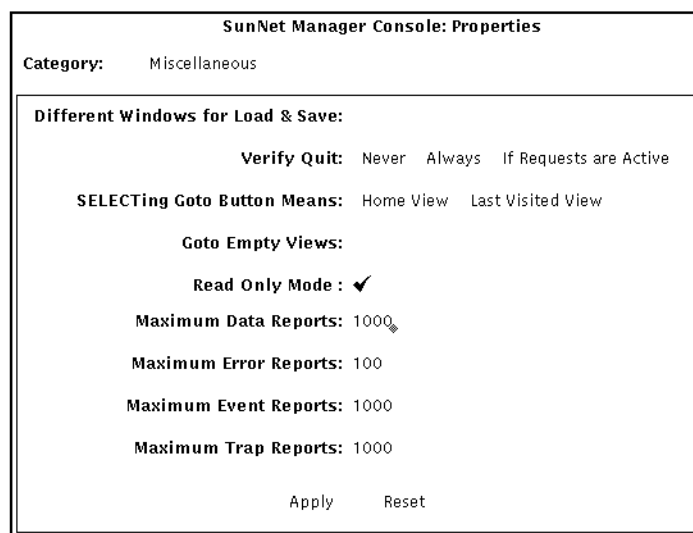


Figure 14-1 Console Read-Only Mode

Note – When operating in this mode, applications such as IP/IPX Discover and CC Receive will not be able to add, modify, or delete topology objects from the runtime database.

14.3 Control Panel Buttons and Menus

The Console base menu contains the following control panel buttons:

- File
- View
- Edit
- Props
- Requests
- Tools
- Goto

They are discussed in the following sections.

14.3.1 File Button

Use the File button menu to load (read) management database (MDB) or schema files, and predefined request records. You can also use this menu to save predefined requests, or instances in the runtime database to a file.

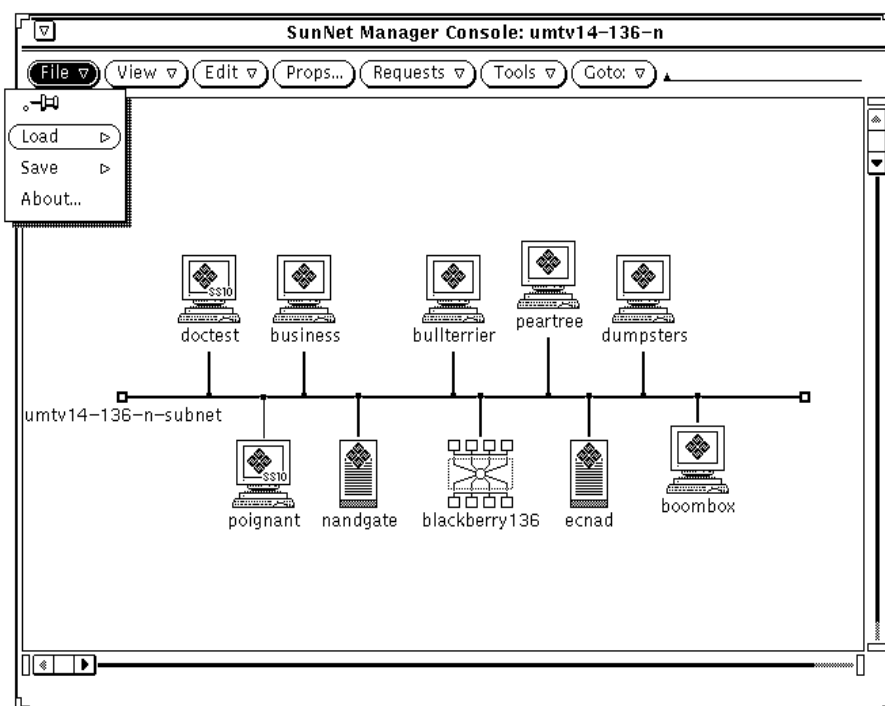


Figure 14-2 File Menu

14.3.1.1 Load Option

Use the Load option to:

- Specify an ASCII-format database file to load into the runtime database.
- Load a predefined data and/or event request file into the runtime database.

14.3.1.2 Save Option

Use the Save option to:

- Save the runtime database version of the predefined data and event request file into an ASCII-format file.

- Save the runtime database into an ASCII-format file.

14.3.1.3 About Option

Use the About option to:

- Find out information about the current release of the product, including version number, licensing requirements, software packages, and documentation titles.

14.3.1.4 Load/Save Management Database Options

The Load and Save options' Management Database window is shown in Figure 14-3.

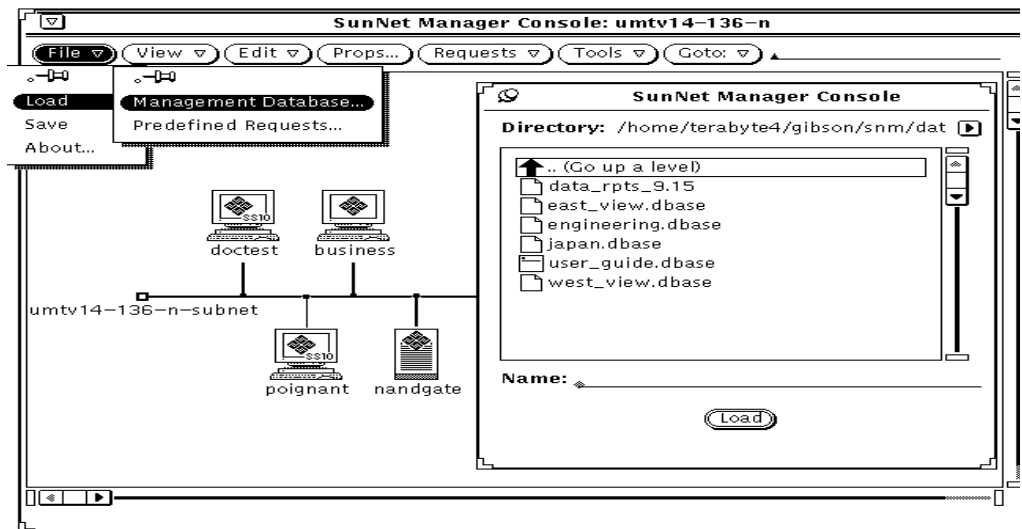


Figure 14-3 Load/Save Management Database Window

The directory from which you started the Console is supplied as the default directory in the Directory field, and its contents are displayed in the scrolling list. To load a file from the list, double-click SELECT on the file name or click SELECT on the file name. Then click SELECT on the Load button. In the Name field, specify a file name in the default directory, a fully-qualified file name, or

an alternate directory path. If you specify a directory path, the directory becomes the default directory, and its contents are displayed in the scrolling list.

If the parser detects an error (for example, incorrect syntax) while loading an MDB file, it displays error messages in the Load/Save window footer and backs out all changes made to the runtime database for the particular file. You must fix and reload the entire file.

Files Must be ASCII

Files must be ASCII-format (for example, a runtime database saved using the File>Save>Management Database option). If you attempt to load a non-ASCII file, an error message is displayed in the Console window footer: *<file name>*: Not Readable by SunNet Manager...

Saving the Current Runtime Database

You can use the File>Save>Management Database option to save the current version of the Console's runtime database to a file. This is especially useful if you have made modifications to the Console and you wish to save the modified Console's runtime database for future use. If, for instance, the machine on which the Console is running goes down and you have saved the runtime database to a file using the File>Save>Management Database option, you can reinvoke SunNet Manager and either specify the saved runtime database file using the *-i* option, or you can use the File>Load>Management Database option to load the modified version you saved earlier.

Unloading the Runtime Database

There is no way to unload the runtime database. To use another database, you must quit the Console session, then restart the Console with the *-i* option.

14.3.1.5 Load Predefined Requests Option

Select the File>Load>Predefined Requests menu option to load a file containing predefined data and/or event request records into the runtime database. This option is intended for use by knowledgeable users who have a file containing their own predefined data and event requests records which they prefer to use instead of, or in conjunction with, the predefined data and event requests provided with the current product. Refer to the Section below,

“Predefined Requests Supplied with SunNet Manager” for more detailed information about the predefined data and event requests provided with the current product. Refer to the Section below, “Creating, Modifying, or Deleting Predefined Requests” for a complete description on how to build, modify, or delete predefined data and event requests.

Requests Must be ASCII Files

To load files containing predefined event or data requests, these files must be ASCII-format files. For example, predefined requests that are saved using the File>Save>Predefined Requests menu option are saved in ASCII format. If you attempt to load a non-ASCII file, an error message is displayed in the footer of the Console window:

```
<file name>: Not Readable by SunNet Manager...
```

Warning – Attempting to load a file containing predefined data and/or event request records that have duplicate names of records already existing in the runtime database will cause the “Load of <filename> failed -- see error report for detailed error message. You can view the error report using the View> menu option. If this problem occurs, use your favorite text editor to remove or rename the duplicate predefined data and/or event record name(s) from the file being loaded. Then, use the File>Load>Predefined Requests menu option

14.3.1.6 Save Predefined Requests Option

Select the File>Save>Predefined Requests menu option to save the runtime database version of the predefined data and event requests records to the ASCII file you specify. Refer to the Section below, “Predefined Requests Supplied with SunNet Manager” for a complete description of predefined data and event requests.

14.4 View Button

Use the View button menu to:

- Display a list of the alarm reports received.
- Display a log of the data reports received.
- Display a log of the event/trap reports received.

- Display a log of the error reports received.
- Display a list of machines that have had an event occur on them.
- Specify a background image to be displayed for the current view.
- Remove a background image from the current view.
- Find a particular element.
- Look at the contents of the clipboard.

The View Console button is described in detail in Chapter 16, “View Reports.”

14.5 Edit Button

The Edit button menu provides a graphical editing capability for creating/deleting, and cutting/copying/pasting glyphs. To access the Edit menu, move the mouse pointer to either the Edit button, or to any open area (where there is no glyph) in the view, and press MENU.

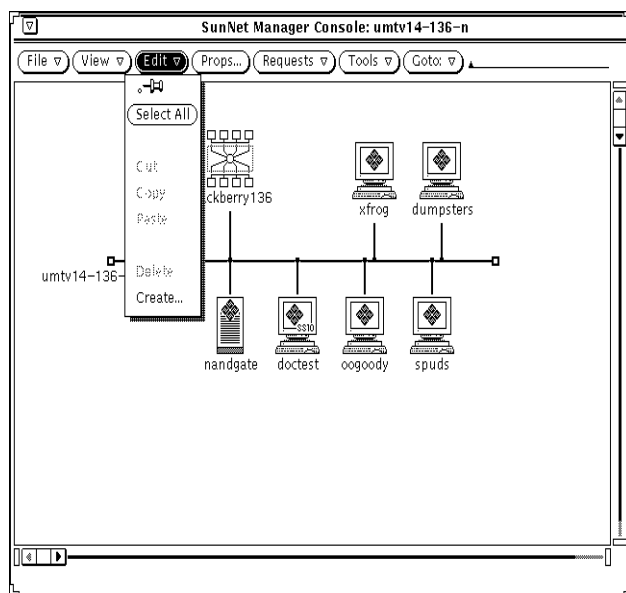


Figure 14-4 Edit Button Menu

Edit operations make use of a clipboard. The clipboard holds elements edited via the SunNet Manager Console graphical editor. The clipboard is read-only — you cannot modify clipboard elements.

14.5.1 *Select All*

Select All selects all elements in the current view for copying, cutting, or deleting.

14.5.2 *Cut*

Cut removes the selected elements from the view and places them in the clipboard. As an alternative, you can press the Cut function key. You can only cut an element if it has a non-empty subview.

14.5.3 *Copy*

Copy copies the selected element(s)—element or request instances—to the clipboard. As an alternative, you can press the Copy function key.

Caution – Perform only one Copy operation at a time. You cannot append to the clipboard—the latest Copy overwrites the clipboard’s previous contents.

14.5.4 *Paste*

Paste inserts the clipboard contents at the current mouse location. Alternatively, you can press the Paste (also known as Get) function key.

You cannot paste an element (a component, view, connection, or bus) in a view where it already is displayed. Duplicate element names are not allowed in a single view.

You can copy and paste a request in a different element’s subview. (See Section 14.10.8, “Show Subview,” on page 14-25 for a discussion of how to navigate to different views using the Glyph►Show Subview menu entry.) Pasting a request into the subview of a particular element inserts the request into that subview and also launches the request targeted at that element. You can use this method to launch the same requests for multiple elements.

To copy elements into a view, you can do one of the following:

- Select one or more elements, Copy them, then go to the target view and Paste.
- Select one or more elements, Copy them, then point at the glyph for the target view and press the Paste function key.
- Move the mouse pointer to the desired element and press SELECT. To copy additional elements, move the mouse pointer to them and press ADJUST. Drag the mouse pointer to the target view. For a single element, release SELECT to place it in the target view. For multiple elements, release ADJUST.

Caution – Perform only one Cut operation at a time. You cannot append to the clipboard—the latest Cut overwrites the clipboard’s previous contents.

14.5.5 Delete

Delete removes the selected element from the view but does *not* place it in the clipboard. Deleting an element causes the record for that element to be removed from the runtime database. You cannot Delete an element that has an active data or event request associated with it. You also cannot Delete an element that has a non-empty subview.

14.5.6 Create

Create displays a window that allows you to create a new element (a component, view, connection, or bus). See Figure 14-5.

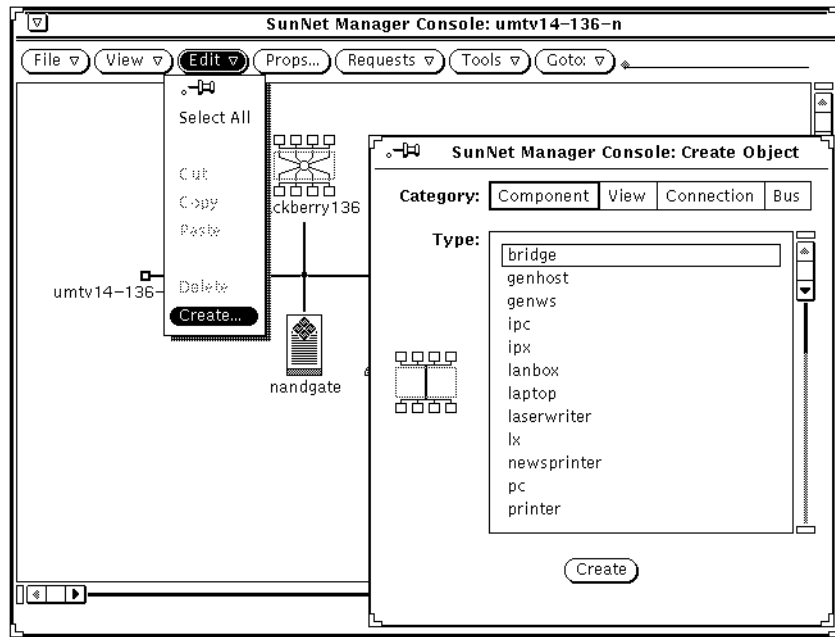


Figure 14-5 Edit>Create Menu

You specify the Category and Type. The glyph image will be displayed for you to preview. When you are satisfied with your selections, click on the Create button. A Properties window for the selected element type appears, as shown in Figure 14-6.

SunNet Manager Console: New component.IPX_Bridge

Name:

IP Address1:

IP Address2:

Contact:

User:

Location:

Description:

Host-Resources	Host Resources
RFC1213-MIB	RFC1213-MIB agent
RFC1229-MIB	RFC1229-MIB agent
RFC1230-MIB	RFC1230-MIB agent
RFC1231-MIB	RFC1231-MIB agent
RFC1232-MIB	RFC1232-MIB agent
RFC1233-MIB	RFC1233-MIB agent
RFC1253-MIB	RFC1253-MIB agent
RFC1271-MIB	RFC1271-MIB agent
RFC1289-nhivM	RFC1289-nhivMIB agent

Red: 0

Green: 0

Blue: 0

Apply Reset Alias...

Create

Figure 14-6 Edit>Create >Element Properties Window

See Section 14.6, “Props Button,” for a description of the Element Properties window. The Alias button at the bottom of the Properties window is dimmed when using Edit>Create to create the new element.

The glyphs in Table 14-1, “Glyphs Used for Multiple Element Types,” are used for more than one type of element.

Table 14-1 Glyphs Used for Multiple Element Types

Glyph	Element Category	Element Type
	Component	ss330 ss370 sun-deskside
	Component	sun-server sun470
	Connection	link rs232
	Component	IPX-Latern Device IPX-PC Unixware
	Component	IPX-Bridge IPX-Router Netware Router
	Component	IPX Print Server IPX Printer

If you use the Edit button to create an element, the glyph is positioned after any explicitly positioned glyphs. Otherwise, the glyph is positioned as near as possible to the upper left-hand corner of the view window. You may also

access the Edit functions by pressing MENU while pointing at any blank space in a view. Creating a new element in this manner positions the glyph at the pointer location. (This is an explicitly positioned glyph.)

Table 14-2 shows a summary of whether an operation can (yes) or cannot (no) be done on the various elements.

Table 14-2 Summary of Edit Operations

Operation	Component/ Bus/View	Connection	Simple Connect	Request
Simple Connect	yes	no	no	no
Copy	yes	yes	no	yes
Create	yes	yes	no	no
Cut	yes	yes	yes	yes
Delete	yes	yes	yes	yes
Double Click	ye	yes	no	no
Drag	yes	yes	no	yes
Drop into	yes	yes	no	no
Paste	yes	yes	no	yes
Properties	yes	yes	no	yes

Note that “simple” connects have not yet been described; see Section 14.10.10, “Connect,” for more information.

14.6 Props Button

The Props button allows you to view, modify, and configure specific element or Console properties. See Chapter 17, “Props Menu,” for a description of the functions of the Console Properties window.

14.7 Requests Button

The Requests button menu allows you to send a Quick Dump request, send data and event requests, build predefined data and event requests, or get a summary of

requests you are sending. See Chapter 15, “Requests Management,” for detailed information about sending requests.

14.8 Tools Button

The Tools button menu allows you to invoke SunNet Manager utilities from the Console. When you click MENU on the Tools button, the menu shown in Figure 14-7, is displayed. Features of this button are discussed in detail in Chapter 24, “Set Tool.”

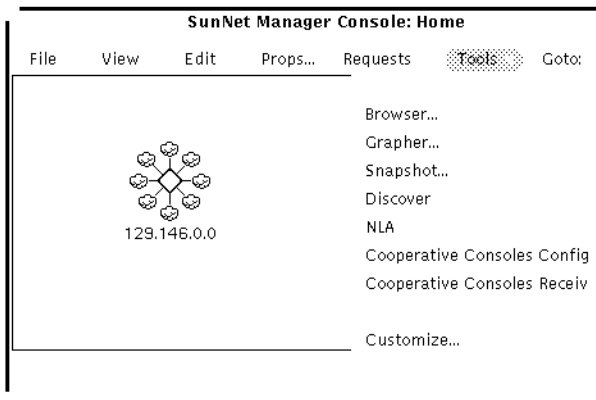


Figure 14-7 Tools Menu

The following options are available through the Tools menu:

- Browser allows you to examine and organize log files. Chapter 20, “Browser,” describes the Browser Tool functions.
- Grapher allows you to visualize data reports and log file information. See Chapter 21, “Results Grapher,” for a description of the Grapher Tool functions.

- Snapshot enables you to capture and print Console screens. For more information about using the OpenWindows Snapshot utility, refer to the *SunOS 4.x DeskSet Environment Reference Guide*.
- IP Discover automatically creates a graphical representation of your network. See Chapter 22, “IP Discover,” for a description of the Discover Tool.
- IPX Discover automatically discovers IPX nodes such as NetWare clients and servers and displays their logical topology. It also discovers NetWare services regardless of protocols used. See Chapter 23, “IPX Discover,” for more information.
- NLA allows you to use the Network Layout Assistant to read the database and automatically place devices and connections. See Chapter 12, “NetWork Layout Assistant,” for more information.
- Cooperative Consoles Config allows you to configure relationships between multiple Site/SunNet/Domain Manager consoles. See the *Cooperative Consoles Administrative Guide* for more information.
- Cooperative Consoles Receive allows you to start the exchange of information between multiple Site/SunNet/Domain Manager consoles. See the *Cooperative Consoles Administrative Guide* for more information.
- Customize allows you to create a customized Tools menu. The customized Tool menu remains in the runtime database, and as such, will carry over to any subsequent invocations of SunNet Manager as long as the `-i` parameter is not specified with the `snm` command.

14.9 Goto Button

The Goto button menu allows you to navigate through views defined in your management database. The Home option displays the home or top-level view. The other options in the Goto menu are the most recently-displayed views. Up to 16 view names in addition to Home can appear in the menu. An example menu is shown in Figure 14-8.

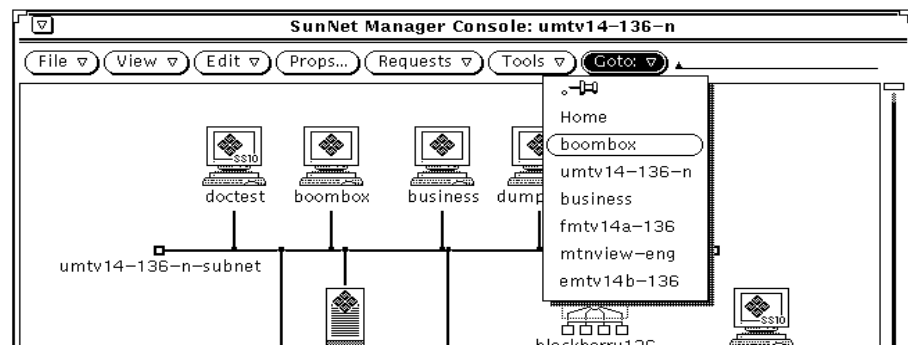


Figure 14-8 Sample Goto Menu

Clicking SELECT on the Goto button causes the last-visited view to be displayed. You can optionally change this operation so that clicking SELECT on the Goto button causes the Home view to be displayed. This is specified in the Miscellaneous category of the Console Properties window. (See Chapter 17, "Props Menu," for more information.) If you know the name of the view you want to see, you can enter the view name on the line next to the Goto button and press Return.

14.10 Element Glyph Menu

The element Glyph menu allows you to:

- Launch a Quick Dump request for an element.
- Launch standard and predefined data and event requests.
- Start a user-defined command on an element.
- Analyze event/trap and error reports.
- Invoke SunNet Manager tools.
- Change the glyph state of an element.
- Display the subview of an element.

- Display and modify the properties of an element.
- Connect an element to another element.
- Change the element type of a component.
- Turn off automatic node management for a subview, if automatic node management is enabled in the Console Properties window. See “Automatic Management” in Chapter 17, “Props Menu,” for more information.

Press MENU on the target glyph to display the Glyph menu as shown in Figure 14-9.

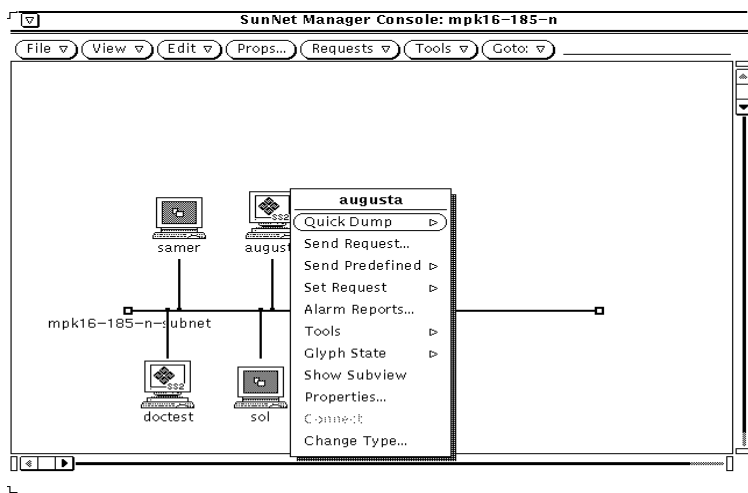


Figure 14-9 Glyph Menu

Following is a description of each item in the glyph menu.

14.10.1 Quick Dump

Quick Dump launches a request for a one-time data report of all the attributes in an agent group. Refer to “Using Quick Dump to Send a Data Report” in Chapter 15, “Requests Management,” for information about using the Quick Dump option.

14.10.2 Send Request

Send Request allows you to send a data or event request. Refer to “Sending Data and Event Requests” in Chapter 15, “Requests Management,” for information about using the Send Request option.

14.10.3 Send Predefined

Send Predefined allows you to send a previously created predefined data or event request. Predefined data and event requests are built by using the SNM Console Requests ► Create Predefined option and assigning a Request Name to the request being built. Refer to “Sending Predefined Data and Event Requests” in Chapter 15, “Requests Management,” for information about using the Send Predefined option.

14.10.4 Set Request

Set Request allows you to set attribute values. The Set Request selection is dimmed if there are no writable attributes for any agent on the target system. Refer to Chapter 24, “Set Tool,” for a detailed discussion.

14.10.5 Alarm Reports

Alarm Reports allows you to analyze event/trap and error reports for a specific element. For a detailed description of the Alarm Reports option, see the discussion under “Viewing Alarm Reports” in Chapter 16, “View Reports.”

14.10.6 Tools

The Tools submenu allows you to execute user-designated commands. See Figure 14-10.

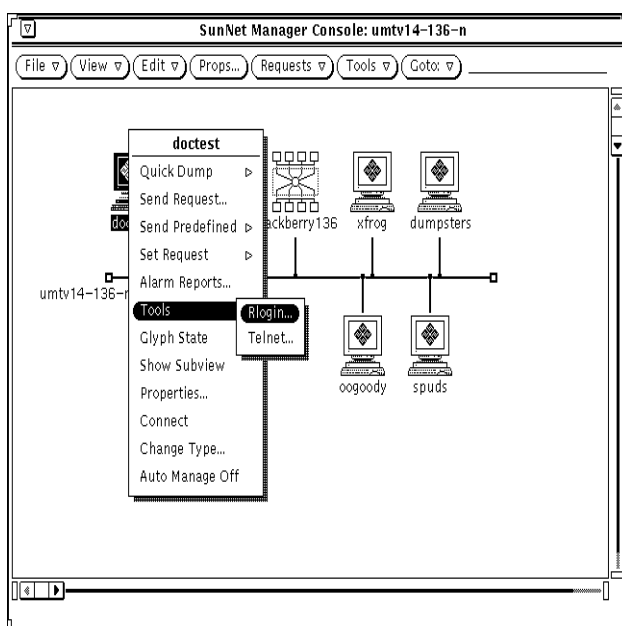


Figure 14-10 Glyph Menu—Tools

The commands in this menu are defined in the `elements.schema` file. (Refer to Chapter 18, “Management Database,” for more information.) The following tools are defined by default:

- **Rlogin**—allows you to log in to the selected machine. This command is defined for workstations only.
- **Telnet**—allows you to communicate with the selected machine, using the TELNET protocol. This command is defined for all devices. However, it may not be valid for devices such as bridges and LAN boxes.

14.10.7 Glyph State

14.10.7.1 *Blinking and Dimming*

Glyph State allows you to turn blinking or dimming on/off for a glyph or turn a glyph that has changed color back to its normal color (see Figure 14-11). For example, you can turn blinking off or turn a glyph back to its normal color once you have recognized that an event has been reported for a particular element. By default, glyph state changes are propagated through the Console's view hierarchy. Refer to the Section on "Events and Traps" in Chapter 17, "Props Menu," for more information about glyph state propagation. Refer to your *SunNet Manager User's Guide for Solaris 2.x/x86* for information on how to have your link(s) blink and turn a different color when an event has occurred.

14.10.7.2 *Pending State*

Starting with version 2.3 of SunNet Manager, a glyph can be put into pending state. In pending state, the request action is cleared and the object is put into a "frozen" state so that no new event or trap for the device will change the state of the glyph. Figure 14-11 shows the "pending state" option. For more information, refer to the Section on "Events and Traps" in Chapter 17, "Props Menu."

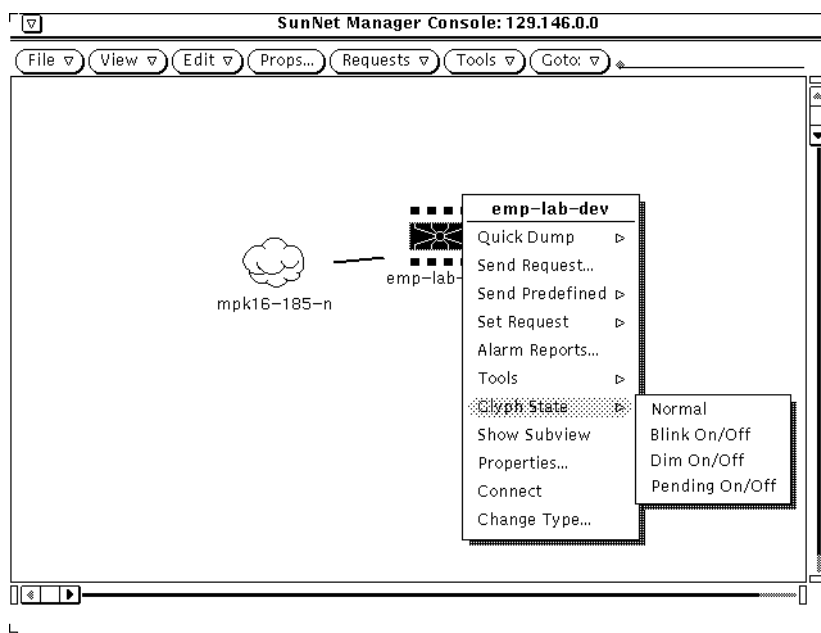


Figure 14-11 Glyph Menu—Glyph States

14.10.8 Show Subview

Show Subview changes views to the next lower level view in the containment hierarchy. The particular subview depends on the context of the current view. The Show Subview option for a subnet shows the element instances that are members of that view. Show Subview for a particular element instance shows all requests targeted at that element.

You can also display an element's view by double-clicking SELECT on the element's glyph. Figure 14-12 shows a sample Show Subview window.

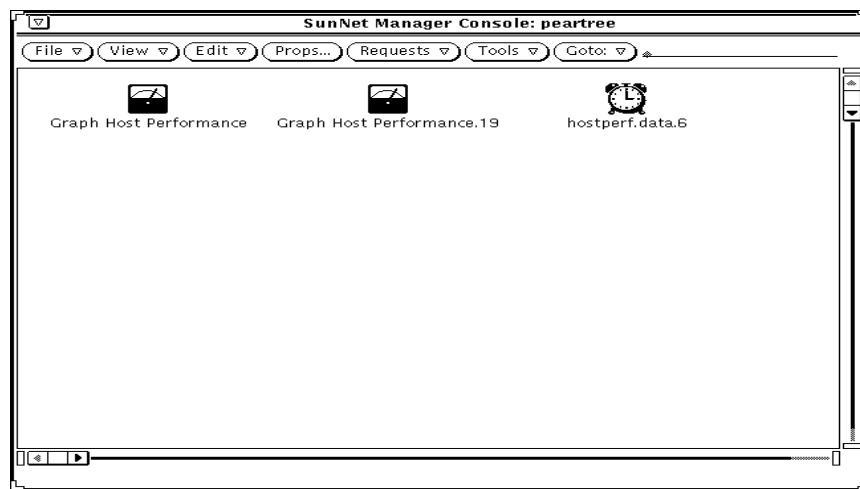


Figure 14-12 Sample Show Subview Window

14.10.9 Properties

The Properties item of the Glyph Menu displays the properties of a particular element glyph.

The Properties pop-up window is divided into four sections: element data, agent/proxy list, color, and a control panel.

- The element data consists of panel text items representing the fields in the element type data record as defined in the `elements.schema` file. These fields are displayed in a scrollable panel.

The values in the fields SNMP RdCommunity (read), SNMP WrCommunity (write), SNMP Vendor Proxy, and SNMP Timeout are parameters that are used only for requests to the SNMP proxy agent. (Refer to Chapter 19, "SNMP Support," for more information about the SNMP proxy agent.) If you do not specify values in the SNMP RdCommunity, SNMP WrCommunity, and SNMP Timeout fields, the SNMP proxy agent uses the following values:

- Read community is “public”
- Write community is “public”
- Timeout is the value (in seconds) specified by the keyword `na.snmp.request_timeout` in the `snm.conf` file on the system where the SNMP proxy agent resides. The keyword’s supplied value is 5 (seconds).

SNMP Vendor Proxy is an optional field that specifies the name of a proxy system with which the SNMP proxy agent will communicate. If this field is specified, the SNMP request is passed through the element to the secondary proxy. This field should only be specified when a vendor has supplied an SNMP proxy agent to manage a particular device or set of devices. The vendor’s SNMP proxy agent communicates with the SunNet Manager SNMP proxy agent via SNMP, but communicates with the element using either SNMP or a different protocol.

Glyph State is a field that appears for an element of types ‘view’ or ‘bus’. This field allows you to specify whether the glyph states of elements in the subview are propagated to the element.

Default Proxy is a field that appears in subnets that correspond to the element category ‘bus’ and element type ‘ethernet’. This field specifies the default proxy system for any elements that are discovered in the subnet. When the Discover Tool is run for the subnet, the name specified in this field appears in the proxy system fields in the Properties window for each element in the subnet. The name you specify for Default Proxy should be the host name of a system on the subnet. This would confine polling and data gathering to the subnet and only relevant management information would pass from the proxy to the Console.

Starting with version 2.3, two attributes have been added:

`Physical_Address` and `SNMP_SysObjectID`.

`Physical_Address` is the MAC address of the device, represented in hex notation. `SNMP_SysObjectID` is a SNMP unique system identifier, represented as an object identifier string. These are read-only fields updated by `IP_discovery`.

For multiport and multi-interface devices, the MAC address of the first two interfaces would be stored. The following components contain two physical address entries (`Physical_Address1`, `Physical_Address2`):

- `component.bridge`

- `component.router`
- `component.hub`
- The agent schema list—agent schema file name, proxy system (if applicable), and brief description—shows all the agent schemas the SunNet Manager Console knows about.

Note – Most of the agents supplied with SunNet Manager have only one agent schema file. The SNMP proxy agent can use Sun-supplied SNMP schema files and/or device-specific schema files created from a MIB file.

The “check” box to the left of the agent schema list provides a convenient method of indicating that this element can be managed using a particular schema. To toggle the check box state between checked and not checked, click SELECT on the check box.

Note – Merely checking an agent schema box does not make the element manageable through the particular agent. The appropriate agent software must be installed and running on the system.

In the case of a proxy agent, the SunNet Manager Console allows you to specify the name of the Sun workstation or server where the proxy agent is running. This proxy system is used as the default in Quick Dump, data, and event reporting requests. If you used the Discover tool to create your management database, each discovered SunNet Manager host is the default proxy system for agents that are installed on the host. For discovered systems that are not SunNet Manager hosts, `localhost`—the Console system—is the default proxy system.

The agent schema list is generated from files in the directories specified by the Schema Locations setting in the Console Properties Locations category (For more information, refer to “Locations” in Chapter 17, “Props Menu.”) If you don’t want an agent schema to appear in the list, remove the corresponding file from the directory. (Before you remove an agent schema file, make sure that the schema is not defined for any elements in an existing instance file. Otherwise, you will not be able to load the instance file.)

-
- The color section of the Properties window provides three fields--red, green and blue--that specify the hue of the glyph associated with a particular machine. To change a hue, press SELECT on the color slider and drag the mouse pointer to the right or the left. Or, enter in numeric values. The intensity for each of the three colors falls in the range of 0 through 255. All zeros makes the glyph transparent; the glyph appears the same color as the background of the Console window. White is all 255s. The box to the left of the color fields displays the currently-defined color.

If you are running the SunNet Manager Console on a monochrome monitor, modifying the color palette has no effect on the glyphs displayed on your machine. However, these modifications are kept in the runtime database. If you save your runtime database to an instance file and use this instance file when running the Console on a color workstation, the color palette values would be used.

- The control panel has the following buttons: Apply, Reset, and Alias. Use the Apply button to put into effect changes made to the properties. Use the Reset button to set the data to its initial state or to the state of the most recent Apply. Use the Alias button to apply one or more alternate names for a machine that has multiple network interfaces, such as a router, for using with SunNet Manager. Figure 14-13 shows the Alias window for the element named `boombox`.

Note – The Alias button will be dimmed if the Edit►Create option was used to create the new element. Refer to Section 14.5, “Edit Button,” for information regarding the Edit►Create option.

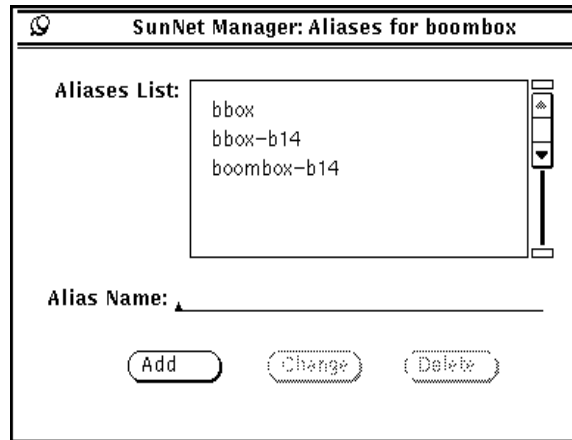


Figure 14-13 Glyph Menu—Properties—Alias Window

Enter a name at the Alias Name prompt and click SELECT on Add to add the name to the Aliases List. To change a name, click SELECT on the name in the Aliases List, make your change to the name as it subsequently appears on the Alias Name line, and click SELECT on Change. To delete a name, click SELECT on the name in the Aliases List, then click SELECT on Delete. At any time, you can click SELECT on the pushpin in the upper right corner of the aliases window to unpin and thereby dismiss the window.

The control panel at the bottom right corner of the element Properties window, indicates if you are in Browse mode (reviewing the properties of an existing element) or Create mode (creating a new element).

Note – Instead of using the Glyph►Properties pull-down menu, you can access the properties of a particular element by moving the mouse pointer over the element glyph and pressing the Props function key.

14.10.10 Connect

Connect draws connections from selected (highlighted) elements to the element glyph on which the mouse arrow is currently pointing. This type of connection is “simple,” in contrast to the connections created through the Edit

menu's Create function. Simple connections are not true elements but are useful for graphically representing connectivity. They can be selected and deleted, but cannot be copied, pasted, or managed. Simple connections inherit the color of the element to which they belong (that is, the connected-to element). Selecting an element also highlights its simple connections; however, connections cannot be moved.

To connect two glyphs, click SELECT on one of them. Move the mouse pointer over the other glyph and choose the Glyph►Connect option. As an example, using two elements called doctest and blackberry136, the connection will be drawn as shown in Figure 14-14 on page 14-31. To delete the connection created by using this feature, click SELECT on the connection, then click SELECT the Edit►Delete option.

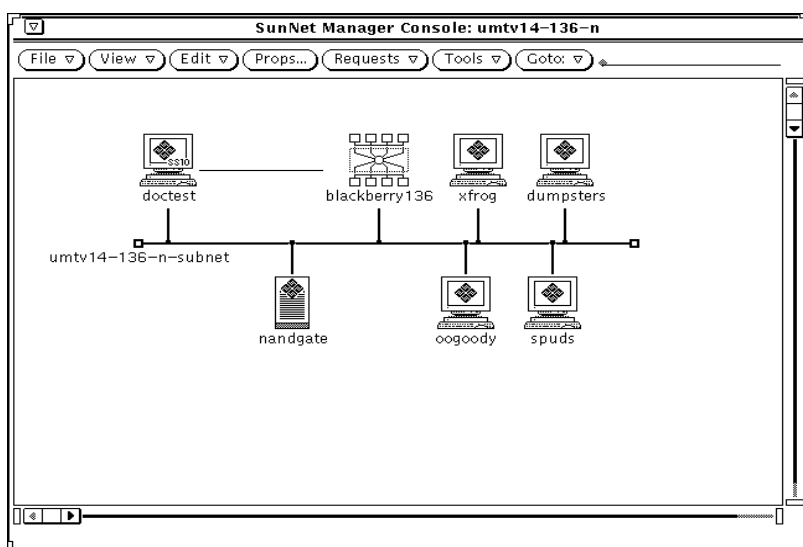


Figure 14-14 Connection Created Between two Elements

14.10.11 Change Type

Change Type allows you to change the element type of a component. Change Type displays a submenu that allows you to select a new component element type. Pull right to the desired element type and release the menu button. The

glyph changes to reflect the glyph that is associated with the new element type. Figure 14-15 shows the Change Type pop-up window that allows you to change an element type.

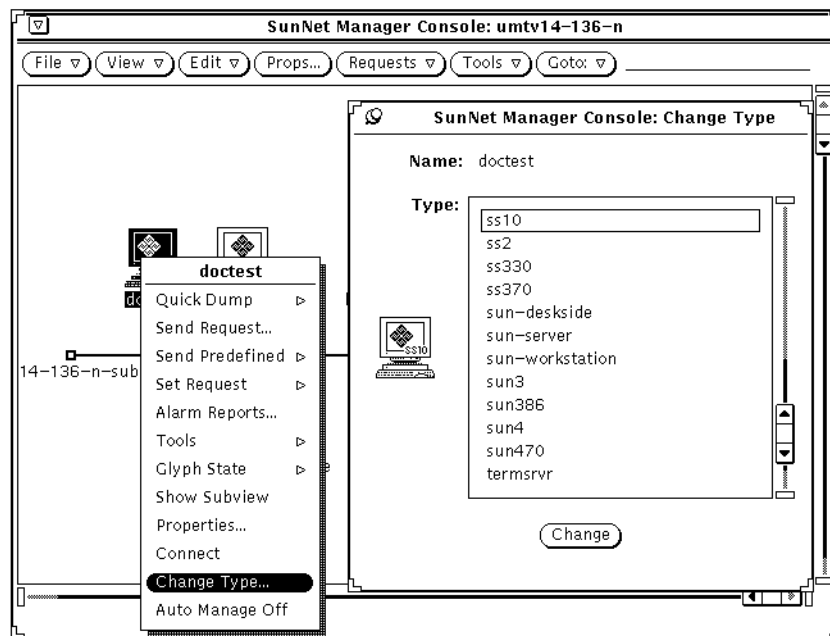


Figure 14-15 Glyph Menu—Change Type Window

When an element type is changed, in some cases not all of the element data fields for the previous element type can be copied over to the new element type. In this case, appropriate warnings are logged in the Error Reports window.

14.10.12 Auto Manage Off

Note – This menu option appears if you enable automatic node management in the Console Properties window. Refer to “Automatic Management” in Chapter 17, “Props Menu,” for more information on this feature.

If automatic node management is enabled, automatic requests are launched for all elements that support at least one of the agents associated with automatic requests. Selecting the Auto Manage Off option, shown in Figure 14-16, specifies that no automatic requests are launched for the element or view. Any active automatic requests for the element or view are killed. Once you choose the Auto Manage Off option, the menu option changes to Auto Manage On. You can use the Auto Manage On option to toggle automatic node management back on for the element or view.

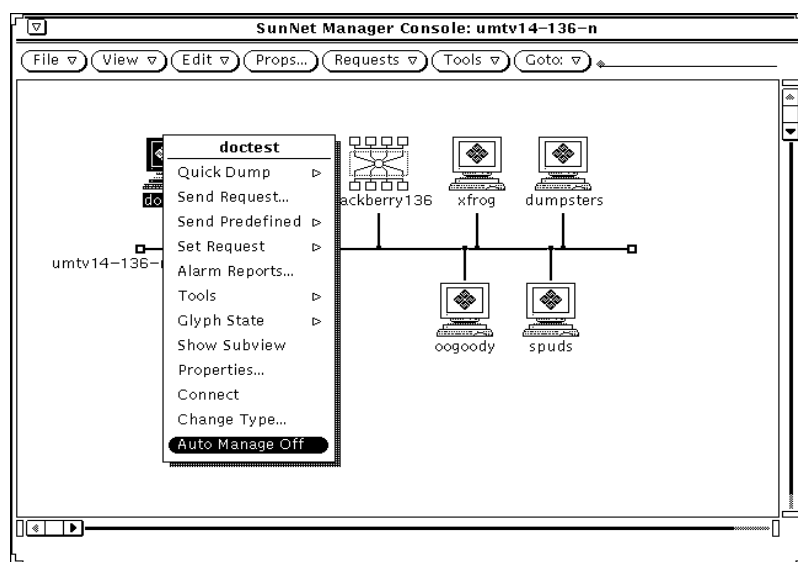


Figure 14-16 Glyph Menu—Auto Manage Off

If the element is a bus or a view, the Auto Manage Off option prevents and kills automatic requests for elements contained in the bus or view. You can toggle automatic management on for individual elements in the view by choosing the Auto Manage On option in the Glyph menu for the element.

Starting with version 2.3 of SunNet Manager, you can use automatic management to launch a predefined event request or chain of requests for specific components. See Chapter 17, "Props Menu," for more information.

Requests Management

15 

This chapter discusses the following topics:

- Quick Dumps through the Request Menu and the Element Glyph Menu
- Send Request option
- Create Predefined Requests
- Request Summary indicator

Use the Request Menu to send a Quick Dump request, send data and event requests, build predefined data and event requests, or get a summary/modify of requests that are being sent. When you press MENU on the Requests button and no elements have been selected, the Requests menu is displayed with the Requests

Summary option as the default option and the Quick Dump and Send Request options dimmed.

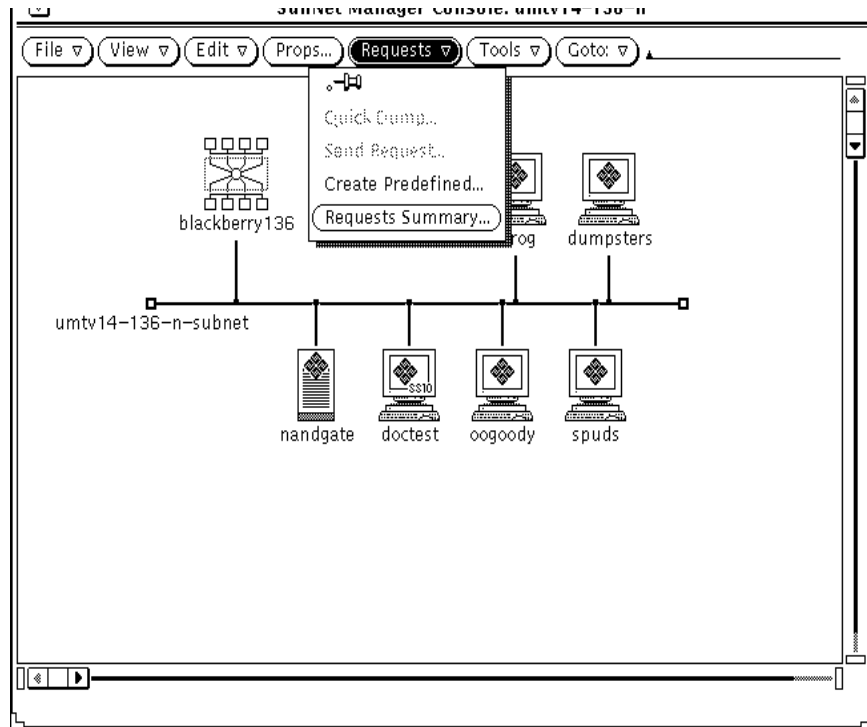


Figure 15-1 Requests Menu

15.0.1 Quick Dump Through the Requests Button

To build a Quick Dump request, select an element then click SELECT on the Requests button and SELECT the Quick Dump menu item. If the Quick Dump menu item in the Requests button menu is dimmed, it means there are no readable attributes for any agent on the target system.

When you have selected the Agent Schema and Group, click SELECT on the Apply button. Use the Reset button to return to the base Quick Dump Request Builder window.

Starting with version 2.3, when you launch a Quick Dump request, you receive a dialog box that tracks the status of the request. For requests with no visible data stream, a timer lets you know the request is processing.

15.0.2 Quick Dump Through the Glyph Menu

Use the Quick Dump option of the element Glyph menu to send a Quick Dump Request for a one-time data report of all attributes in an agent group. If there are no readable attributes for any agent on the target system, the Quick Dump item is dimmed. Figure Figure 15-2 is an example of launching a Quick Dump request from the element flower Glyph menu for the Agent Schema named diskinfo of Group type diskSpace.

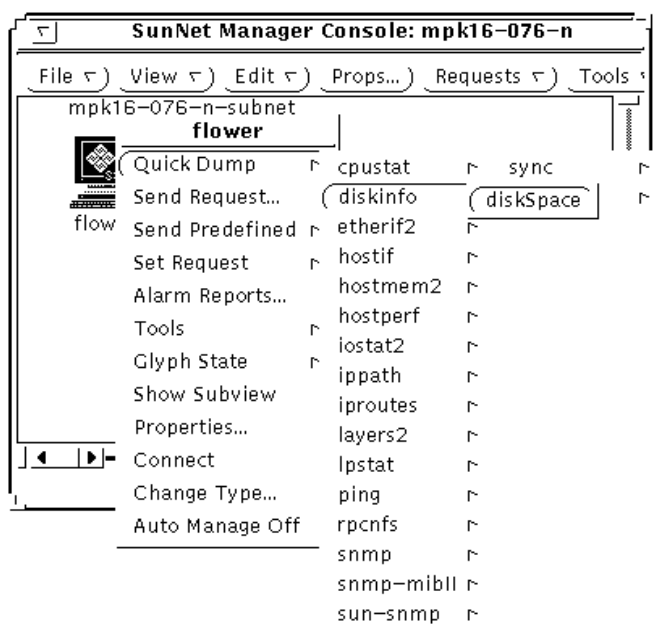


Figure 15-2 Quick Dump Request

15.1 Send Request

The Send Request option is used to send a data or event request from the selected element. The Send Request option is dimmed if an element was not selected prior to selecting the Requests button or if the selected element has no agents specified.

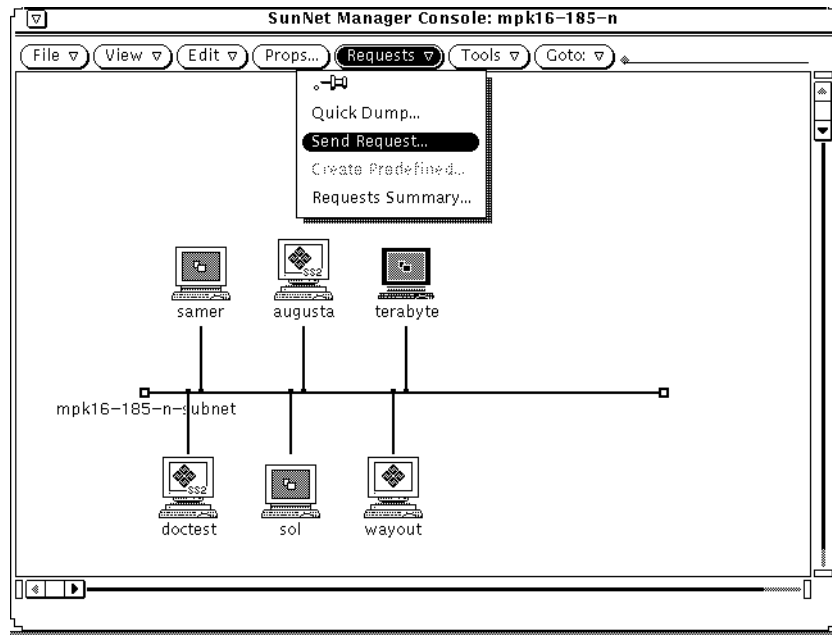


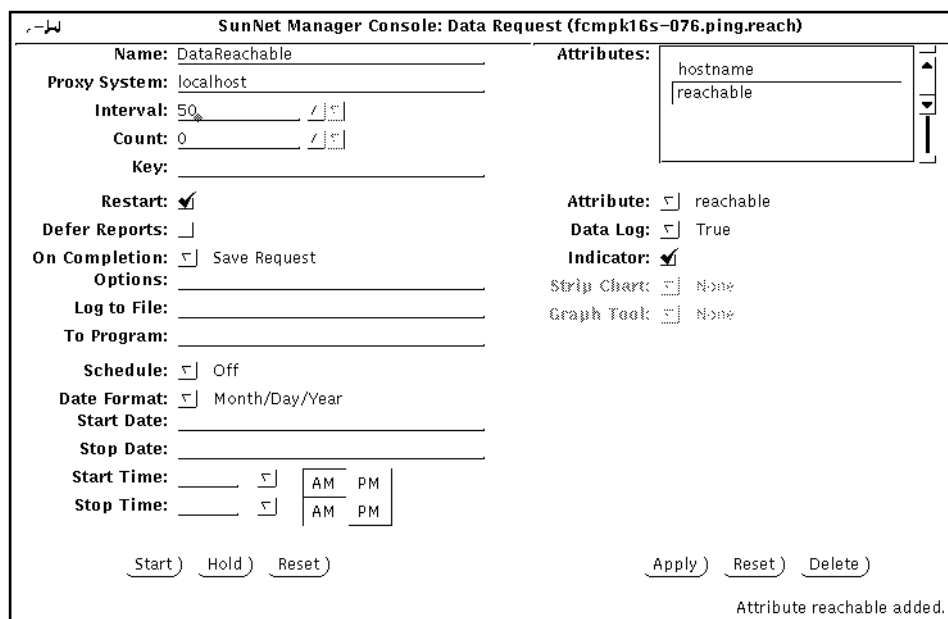
Figure 15-3 Send Requests Window

Data and event requests send information such as the number of reports and reporting intervals to the specified agents on the target machine. The agents automatically generate reports at specified intervals without intervention (polling) from the Console. The activity daemon running on the Console workstation periodically verifies that the specified agent is running on the target machine. In the case of proxy agents, the proxy agent polls the target system at the intervals specified in the data or event request.

See the Tasks section of this Manual for examples of sending data and event requests.

15.1.0.1 Data Request Template

Figure 15-4 is an example of a data request named DataReachable. A description of the fields in the template are provided in Figure 15-4.



Screenshot of the SunNet Manager Console: Data Request (fcmpk16s-076.ping.reach) window. The window displays configuration fields for a data request named DataReachable. The fields are organized into two columns.

Left Column Fields:

- Name: DataReachable
- Proxy System: localhost
- Interval: 50
- Count: 0
- Key:
- Restart:
- Defer Reports:
- On Completion: Save Request
- Options:
- Log to File:
- To Program:
- Schedule: Off
- Date Format: Month/Day/Year
- Start Date:
- Stop Date:
- Start Time: AM PM
- Stop Time: AM PM

Right Column Fields:

- Attributes: hostname, reachable
- Attribute: reachable
- Data Log: True
- Indicator:
- Strip Chart: None
- Graph Tool: None

Buttons: Start, Hold, Reset, Apply, Reset, Delete

Status: Attribute reachable added.

Figure 15-4 Data Request Template

The description of the agent group for the request is displayed in the lower left corner of the window. The properties fields for this data request need to be specified at this time. Each field is described below.

Reporting Characteristics

The fields on the left side of the Data Request window specify reporting characteristics. Each field is described below.

Name:

The optional name you wish to assign to the request. This allows you to assign a recognizable name to a request. The request manager assigns a request name, if one is not specified, using the following format:

`<agent>.<group>.<number>` (for example, `hostperf.data.0`)

Proxy System:

The name of the system the data request is to be sent to.

Interval:

Specifies the interval (in seconds) for the agent on the target system to send reports. An Interval value of 0 indicates that the agent should use its default interval. See the agent's `man` page for its default interval.

Count:

Specifies the number of times for the agent on the target system is to send reports. A count of 0 specifies that the agent is to send reports until the request is stopped or killed. If a count is specified, and a Stop Date/Time is/are specified, the request stops if the specified number of reports are generated before the Stop Date/Time are reached. If the Stop Date/Time is/are reached before the specified number of reports are generated, the request stops at the specified time.

Key:

Identifies a particular row in a table (for example, `ie0` or `le0` in an Ethernet interface table). The use of a key is agent-specific and is indicated in the bottom left corner of the Data Request Properties window. See the `man` page for the particular agent for information regarding keys. If no key is specified, the entire table is returned.

Restart:

Specifies whether the agent should attempt to restart the request if the system on which the agent is running reboots, the agent terminates unexpectedly, or the Console itself is restarted. If Restart is off (no check mark appears in the accompanying box), the request is discarded if the agent fails or if you quit the Console. If Restart is on (a check mark appears in the accompanying box), the request is restarted. The default setting of this field is determined by the Restart Request upon Agent Failure setting in the Console Properties Requests category. Click SELECT on the box next to this field to toggle the check mark on or off.

Defer Reports:

Specifies whether or not the agent should cache reports and send them to the Console only when asked to. If Defer Reports is on (a check mark appears in the accompanying box), the agent is directed to collect the statistics, but send them to the Console only when directed. Defer Reports

tells the agent to cache the last 32 reports. If 32 reports are cached, or if the agent runs out of memory, the oldest report is deleted when a new report is cached. You should not specify Defer Reports for agents that return a large amount of data (for example, a routing table).

In many cases an agent collects information useful for debugging problems. This information might not be of interest in your normal daily operation. Furthermore, if a request were started before the error occurred, data reports will continually stream back to the Console causing unnecessary network traffic and increased CPU load. If, on the other hand, the request was started after the error happened, the debugging information would not have been collected. By setting Deferred Sending on, reports are held in the agent's system until you need to ask for them.

The data request must be active for Defer Reports to work properly; that is, issue the request with a long Interval and high Count. Refer to the “Viewing and Modifying Requests” section for more information on how to obtain deferred reports.

On Completion:

Specifies whether the request should be deleted or saved upon completion. To specify that the request be saved, drag MENU over the On Completion abbreviated menu button—release MENU over Save Request. If you choose to save the request, the request glyph is dimmed after the request has been completed but remains in the view of the target element. You can examine, modify, or restart the saved request.

If you choose Delete Request on Completion, stopping the request is the same as killing it. (Refer to the “Viewing and Modifying Requests” section for more information.)

Options:

Specifies any options (such as arguments) that the agent expects. To specify Options, type in an option string. For example, with SNMP requests you can specify an SNMP read-community name. The information that you can specify in the Options field is agent-specific—not all agents accept options. See the `man` page for the particular agent for information regarding options.

Note – Starting with version 2.3, there is a new flag called `get-requested-attribute-only`. By default it is set to true. Only the requested attributes will be retrieved. When this flag is set to false, all attributes are retrieved.

Log to File:

Specifies the file name where reports are to be stored. If you specify a file name with no path, the Console uses the current directory. If you fully qualify the file name and the left-most characters scroll left, they are not lost.

To Program:

A shell command line specifying a program or shell script to be run. Include the directory path, if necessary. Data reports are passed to the standard input of the program or shell script. Every active program to which data reports are sent keeps a file descriptor open. The default maximum number of file descriptors allowed for a single process is 64. You can increase this number up to 256 by using the limit command in the C-shell.

Schedule:

Set this field to “On” in order to activate the scheduling options below.

Date Format:

Select one of the four formats in which you will enter the start and stop dates.

Start Date:

When you enter a date here, and click the Start button, the request will start on the specified date.

Stop Date:

When you enter a date here, the request will stop on the specified date.

Start Time:

When you enter a time here, and click the Start button, the request will start at the specified time.

Stop Time:

When you enter a time here, the request will stop at the specified time.

Selecting Attributes

The fields in the right portion of the Data Request window are used to specify attributes.

Attributes:

A scrolling list that contains those attributes for which you are requesting data. You choose an attribute and how you want the attribute data displayed with the fields beneath the Attributes scrolling list (described below). When you click SELECT on the Apply button at the bottom right of the window, the attribute is added to the Attributes scrolling list.

Following is a description of the Attribute and attribute display fields:

Attribute:

Specifies the attribute you want the agent to return data on. When the Data Request window is first displayed, the Attribute field displays the name of the first attribute in the group/table from the agent schema file. To specify a particular Attribute, press MENU on the Attribute abbreviated menu button and release MENU on the desired entry. When an attribute is selected, a description is displayed in the lower-right footer.

Note – If no attribute is specified, *all* attributes for the selected group or table are written to the Data Reports log. For tables, the values of the keys are also returned. However, if you specify a particular table attribute, the key values are not returned.

Data Log:

Specifies whether or not the attribute data should be written in the Data Reports Log. To specify that the data *not* be written in the Data Reports log, drag MENU over the Data Log abbreviated menu button—release MENU over False.

Indicator:

Specifies whether or not the attribute data should be displayed in an Indicator. An Indicator shows the last reported value for a particular attribute. Refer to the “Indicators” section for more information.

To specify the Indicator display option, click SELECT on the Indicator box.

Strip Chart:

Specifies whether or not the attribute data should be displayed in a Strip Chart. A Strip Chart contains an auto-scaled strip chart of values for an attribute. Refer to the “Strip Charts” section for more information.

To specify the Strip Chart display option, drag MENU over the Strip Chart abbreviated menu button—release MENU over either Absolute Values (chart the received values) or Delta Values (chart the differences between received values).

Graph Tool:

Specifies whether or not the attribute data should be displayed by the Grapher. The Graph Tool, also known as the Results Grapher, is an application that allows the display of data received by the Console. Only data of types integer, float, counter, gauge, timestamp, or UNIX time can be plotted. Refer to Chapter 21, “Results Grapher,” for more information about the Graph Tool. To specify the Graph Tool display option, drag MENU over the Graph Tool abbreviated menu button—release MENU over either Absolute Values (graph the received values) or Delta Values (graph the differences between received values).

Adding Attributes

The buttons below the Attributes fields in the Data Request window allow you to add, change, and delete attribute selections in the Attributes scrolling list. The buttons are described below:

- Apply tells the request manager to add the Attribute selected to the Attributes scrolling list.
- Reset tells the request manager to reset the Attributes scrolling list back to its default value.
- Delete tells the request manager to remove the Attribute selected from the Attributes scrolling list.

An entry in the scrolling list may be selected at any time by pointing the mouse pointer at the entry and clicking the SELECT button. This updates the window to the settings for the selected attribute. An entry may be deleted by selecting it and clicking on the Delete button.

Sending the Data/Event Request

The buttons at the bottom left of the Data/Event Request window allow you to send, hold, and reset the data/event request. The buttons are described below:

- Start tells the request manager to send the data/event request.
- Hold tells the request manager to hold the data/event request.
- Reset tells the request manager to set the data/event request to its default values.

Once you have specified the options that you desire in the Request Builder template, click SELECT on the Apply button. Refer to “Data Request Template,” above for a description of the template that will be displayed next. Again, specify the desired options and then click on the Start button to send the data request to the desired agent.

15.1.0.2 Event Request Template

Figure 15-5 is an example of an event request named `if System Reboot`.

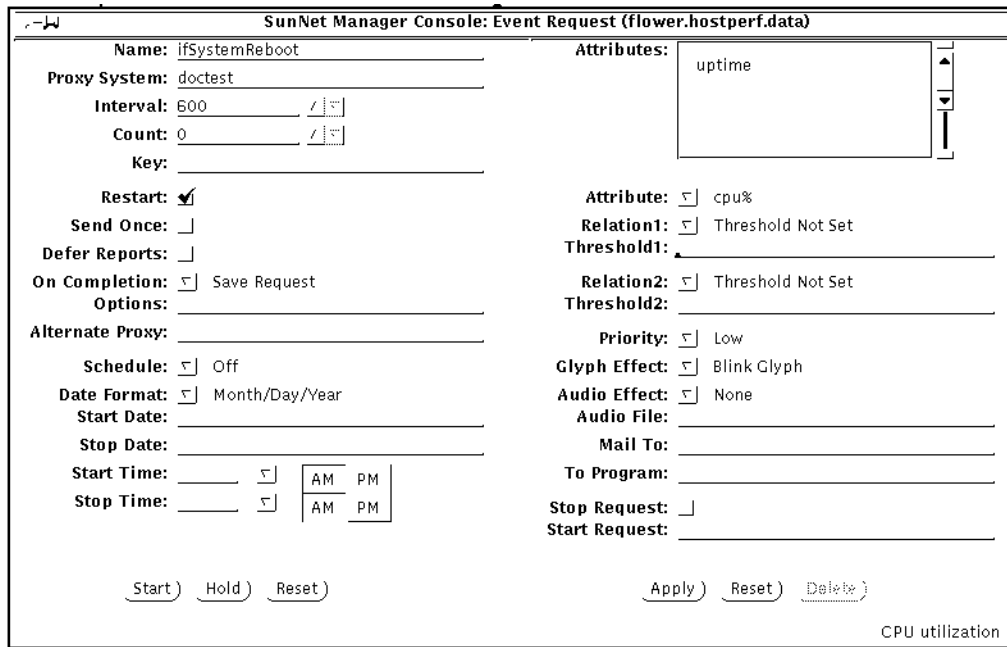


Figure 15-5 Event Request Template

Reporting Characteristics

The fields on the left side of the Event Request window specify reporting characteristics. Each field is described below.

Name:

The optional name you wish to assign to the request. This allows you to assign a recognizable name to a request. The request manager assigns a request name, if one is not specified, using the following format:

<agent>.<group>.<number> (for example, *hostperf.data.0*)

Proxy System:

The name of the system the data request is to be sent to.

Interval:

Specifies the interval (in seconds) for the agent on the target system to send reports. An Interval value of 0 indicates that the agent should use its default interval. See the agent's `man` page for its default interval.

Count:

Specifies the number of times for the agent on the target system is to send reports. A count of 0 specifies that the agent is to send reports until the request is stopped or killed. If a count is specified, and a Stop Date/Time is/are specified, the request stops if the specified number of reports are generated before the Stop Date/Time are reached. If the Stop Date/Time is/are reached before the specified number of reports are generated, the request stops at the specified time.

Key:

Identifies a particular row in a table (for example, `ie0` or `le0` in an Ethernet interface table). The use of a key is agent-specific and is indicated in the bottom left corner of the Data Request Properties window. See the `man` page for the particular agent for information regarding keys. If no key is specified, the entire table is returned.

Restart:

Specifies whether the agent should attempt to restart the request if the system on which the agent is running reboots, the agent terminates unexpectedly, or the Console itself is restarted. If Restart is off (no check mark appears in the accompanying box), the request is discarded if the agent fails or if you quit the Console. If Restart is on (a check mark appears in the accompanying box), the request is restarted. The default setting of this field is determined by the Restart Request upon Agent Failure setting in the Console Properties Requests category. Click SELECT on the box next to this field to toggle the check mark on or off.

Send Once:

If you check this box, the event request is killed after the console receives the event. If the box is not checked, an event report is sent for every event occurrence.

Defer Reports:

Specifies whether or not the agent should cache reports and send them to the Console only when asked to. If Defer Reports is on (a check mark appears in the accompanying box), the agent is directed to collect the statistics, but send them to the Console only when directed. Defer Reports

tells the agent to cache the last 32 reports. If 32 reports are cached, or if the agent runs out of memory, the oldest report is deleted when a new report is cached. You should not specify Defer Reports for agents that return a large amount of data (for example, a routing table).

In many cases an agent collects information useful for debugging problems. This information might not be of interest in your normal daily operation. Furthermore, if a request were started before the error occurred, data reports will continually stream back to the Console causing unnecessary network traffic and increased CPU load. If, on the other hand, the request was started after the error happened, the debugging information would not have been collected. By setting Deferred Sending on, reports are held in the agent's system until you need to ask for them.

The data request must be active for Defer Reports to work properly; that is, issue the request with a long Interval and high Count. Refer to the “Viewing and Modifying Requests” section for more information on how to obtain deferred reports.

On Completion:

Specifies whether the request should be deleted or saved upon completion. To specify that the request be saved, drag MENU over the On Completion abbreviated menu button—release MENU over Save Request. If you choose to save the request, the request glyph is dimmed after the request has been completed but remains in the view of the target element. You can examine, modify, or restart the saved request.

If you choose Delete Request on Completion, stopping the request is the same as killing it. (Refer to the “Viewing and Modifying Requests” section for more information.)

Options:

Specifies any options (such as arguments) that the agent expects. To specify Options, type in an option string. For example, with SNMP requests you can specify an SNMP read-community name. The information that you can specify in the Options field is agent-specific—not all agents accept options. See the `man` page for the particular agent for information regarding options.

Note – Starting with version 2.3, there is a new flag called `get-requested-attribute-only`. By default it is set to true. Only the requested attributes will be retrieved. When this flag is set to false, all attributes are retrieved.

Alternate Proxy

Use this field to specify an alternate proxy you wish to use to launch a request that has previously been launched with a different proxy agent. See “Alternate Proxy” for more information on this feature.

Schedule:

Set this field to “On” in order to activate the scheduling options below.

Date Format:

Select one of the four formats in which you will enter the start and stop dates.

Start Date:

When you enter a date here, and click the Start button, the request will start on the specified date.

Stop Date:

When you enter a date here, the request will stop on the specified date.

Start Time:

When you enter a time here, and click the Start button, the request will start at the specified time.

Stop Time:

When you enter a time here, the request will stop at the specified time.

Important Information About Interval and Count fields

For event requests, interval and count fields simply specify when the agent is to send a report. A report is forwarded to the Console system only if it has been determined that an event has occurred. For agents that are shipped with the product, the value reported for an attribute is the value noted at the reporting interval. Thus, it is possible for event conditions to occur *between* reporting intervals and not cause an event report.

If Send Once has been selected (a check mark appears in the accompanying box), the request is terminated after the Console receives the first event report. Click SELECT on the Send Once box to toggle the check mark on or off.

Alternate Proxy

Starting with version 2.3, you can use an alternate proxy to create an event request against a device. When an event is generated against a device and a value name is specified in the alternate proxy field, the request is started on

this proxy agent on the specified host. This become useful in a troubleshooting a situation such as the following: You are pinging a device in a remote location, and it responds by indicating that the device is down. To verify that it is the device and not, for example, an intermediate router, you can send the same request using an alternate proxy agent. You might then want to try an alternate route through another proxy agent on an alternate network. This way, you can verify the router is down, not the device. The alternate proxy feature enables you to do this verification automatically.

The alternate proxy field is enabled only when you send requests against proxy agents.

Descriptions of the Attribute fields are provided below:

Attributes:

A scrolling list that contains attributes for which you want event information.

Attribute:

Specifies the attributes that you want to use to specify an event. Click SELECT on the button to see the list of available attributes. After you set attribute threshold(s), click SELECT on Apply to add the attribute to the Attributes list.

Relation1, Relation2:

Specify threshold conditions for reporting an event. You can specify up to two threshold relational operators; the event is reported if *either* of the two threshold conditions are met. To set the threshold relational operator(s) for

Relation1 or Relation2, drag MENU over the Relation abbreviated menu button to display the following list of relational operators in a pop-up menu—release MENU over the desired relational operator.

Threshold Not Set means you have not set a threshold for this attribute. This is the default choice.

Equal To tells the agent to report an event if the attribute value equals “Threshold n ”.

Not Equal To tells the agent to report an event if the attribute value does not equal “Threshold n ”.

Less Than tells the agent to report an event if the attribute value is less than “Threshold n .”

Less Than Or Equal To tells the agent to report an event if the attribute value is less than or equal to “Threshold n .”

Greater Than tells the agent to report an event if the attribute value is greater than “Threshold n .”

Greater Than Or Equal To tells the agent to report an event if the attribute value is greater than or equal to “Threshold n .”

Changed tells the agent to report an event if the attribute value differs from the last sampled value.

Increased By tells the agent to report an event if the attribute value has increased by “Threshold n ” compared to the last sampled value.

Decreased By tells the agent to report an event if the attribute value has decreased by “Threshold n ” compared to the last sampled value.

Increased By More Than tells the agent to report an event if the attribute value has increased by more than “Threshold n ” compared to the last sampled value.

Increased By Less Than tells the agent to report an event if the attribute value has increased by less than “Threshold n ” compared to the last sampled value.

Decreased By More Than tells the agent to report an event if the attribute value has decreased by more than “Threshold n ” compared to the last sampled value.

Decreased By Less Than tells the agent to report an event if the attribute

value has decreased by less than “Threshold n ” compared to the last sampled value.

Note – For the Changed, Increased By, Decreased By, Increased By More Than, Increased By Less Than, Decreased By More Than, and Decreased By Less Than operators, the Count field of the report characteristics must not be ‘1.’ A Key must be supplied if a table of attributes is selected.

Threshold1, Threshold2:

Specify the threshold value if the selected operator is other than Threshold Not Set, or Changed.

Priority:

Specifies the priority of the request. Click SELECT on the desired priority button (Low, Medium, or High) for the request.

Glyph Effect:

Specifies a visible indicator when the specified event is reported. Press MENU on the Glyph Effect abbreviated menu button to display a signal options pop-up menu. Release MENU on the signal option you want. You can choose only one visual signal option. In order to have the “decay” feature in effect, you must select the Priority by Color option.

The following options are available:

Blink Glyph

Blinks the glyph of the element for which an event has been reported.

Dim Glyph

Dims the glyph of the element for which an event has been reported.

Priority by Color

Causes the glyph to change color, based on the priority of the reported event. The defaults are:

- red = high priority
- orange = medium priority
- yellow = low priority
- blue = decay

Starting with version 2.3, you can customize colors, including the “decay” color. See Chapter 4, “Requesting Data,” or Chapter 5, “Specifying Event Requests,” for procedures to customize Color by Priority.

If multiple reports are received for a system, the highest priority is used. In association with the Priority by Color option is the “decay” feature. This feature pertains to events and traps that have reached a threshold and then fallen below it, or that have had an event or trap stop being reported. In response to these conditions, the glyph for the affected element turns blue, which is the default color, or to the color you have customized. Once a glyph has decayed to blue, the event must be acknowledged in order for the glyph to return to its original color. To get a glyph to return to its original color, you can either:

- Select the Console’s View►Event Summary option
 - Highlight the name of the glyph
 - Press the Drop from List button.
- Or
- From the Glyph Menu for the specific element, pull right over the Glyph State►Normal option.

Audio Effect:

Specifies an audible indicator when the specified event is reported. Press MENU on the Audio Effect abbreviated menu button to display a signal options pop-up menu—drag and release MENU on the desired signal option. Only one audio signal option can be chosen. The following options are available:

Ring Bell sounds an audible bell on the Console workstation.

Play Audio File causes the file specified in the Audio File field to be played.

Audio File:

Defines the audio file to be played when the specified event is reported. Type in the file name of the audio file. The option Play Audio File must be specified in the Audio Effect field. You can use the volume function of the Audio Tool in OpenWindows 3.2 to control the sound level.

Note – The Console must be running (not merely displaying) on a machine with an audio port. If you are running the Console on a server but displaying the Console windows on a local workstation, the Console will attempt to play the audio file on the server.

Mail To:

Defines one or more mail recipients to send the event report to when the specified event is reported. Type in a list of mail recipients. If there is more than one recipient, use a space between each entry.

To Program:

A shell command line specifying a program or shell script to be run. Include the directory path, if necessary. A new copy of the program or shell script is forked for each event report. The event report is passed to the standard input of the program or shell script. Every active program that receives an event report keeps a file descriptor open. The default maximum number of file descriptors allowed for a single process is 64. You can increase this number up to 256 by using the `limit` command in the C-shell.

Stop Request:

This field is new starting with version 2.3. If this field is checked, the request will be stopped when a specific event is received. This feature enables you to specify a stop request at the attribute level. You can set thresholds against different attributes in an event request and also specify that the request stop only when a *specific attribute* exceeds a threshold's level.

Start Request:

This field is new starting with version 2.3. A new request name (including predefined requests) can be specified in this field. A different request name can be specified for each event attribute. Whenever an event is generated against the given event request, the request specified in this field is launched automatically upon receipt of the event. This can be used in conjunction with the other event based options such as stopping the request or trying an alternate proxy.

15.2 Create Predefined

The Create Predefined option is used to build a predefined data or event request. This is a global operation, not element based. Therefore, this menu item will be dimmed if an element has been selected prior to clicking the Requests button. Refer

to the “Creating, Modifying, or Deleting Predefined Requests” section for information about using the Create Predefined option.

Note – The predefined data and event record names are not case sensitive, as is true with most of the Console related name fields. Since the predefined data and event records reside in the runtime database with all of the elements, it is *strongly* recommended that the names assigned to newly created predefined requests consist of at least two words, as is the case with the predefined data and event requests provided with this product. Otherwise, a conflict may arise between the names of hosts and the names of the predefined data and event request records, as they all must be unique.

15.2.1 *Supplied Predefined Requests*

A number of predefined data and event requests are supplied and recorded in the `.SNMPpredefined` file in the `struct` directory. Upon invoking the Console, a check is made to see if the `$HOME/.SNMPpredefined` file exists. If it does, the file is loaded into the runtime database for use by the Console. If it does not exist, the `.SNMPpredefined` file is loaded into the runtime database. The first time the Console is invoked, the `.SNMPpredefined` file will be loaded into the runtime database. The records in this file, along with any predefined data and event requests you modify or create after the Console is started, are stored in your `$HOME/.SNMPpredefined` file upon exiting from the Console. You can also save the runtime database version of the predefined data and event requests records by using the `File>Save>Predefined Requests` option. Using this option, you can save these records to your `$HOME/.SNMPpredefined` file or to any other file you desire. For information about creating, modifying, deleting, and sending predefined data and event requests, refer to the sections that follow.

15.2.2 Predefined Data Request Records

The following table lists the predefined data requests that are provided with this product, their Agent, Group, and Attribute settings, as well as a description of their intended function(s).

Table 15-1 SNM Supplied Predefined Data Requests

Request Name	Agent Name	Group Name	Attributes Supported	Function
Graph Host Performance	hostperf	data	cpu% intr disk ipkts opkts	CPU utilization # of device interrupts # of disk transfers # of if input pkts # of if output pkts
Record Disk Space	diskinfo	diskSpace	all	disk space information
Record Host Performance	hostperf	data	all	Statistics of Host
Show Host Interfaces	hostif	if	all	host interface statistics
Show NFS Statistics	rpcnfs	client	all	RPC and NFS statistics
Show Path to Host	ippath	path	all	trace IP packet's path between proxy and target system
Show Routing Statistics	iproutes	routes	all	Routing statistics
Show snmp System Info	snmp	system	all	System information
Show snmp-mibII System Info	snmp-mibII	system	all	System information
Show sun-snmp System Info	sun-snmp	system	all	System information

15.2.3 Predefined Event Request Records

The following table lists the predefined event requests that are provided with this product, their Agent, Group, and Attribute settings, as well as a description of their intended function(s).

Table 15-2 SNM Supplied Predefined Event Requests

Request Name	Agent Name	Group Name	Attributes Supported	Event to be Reported
When Disk is Full	diskinfo	diskSpace	capacity	Disk file system is full
If System Reboot	hostperf	data	uptime	If system reboots
When Printer Error	lpstat	status	statusCode	Line printer error
When System is not Reachable	ping	reach	reachable	System not reachable

15.2.4 Predefined Requests Management

A Predefined Data Request Template similar to the one in Figure 15-6 appears after you select the Request Type, Request Name, Agent Schema, and Group Name.

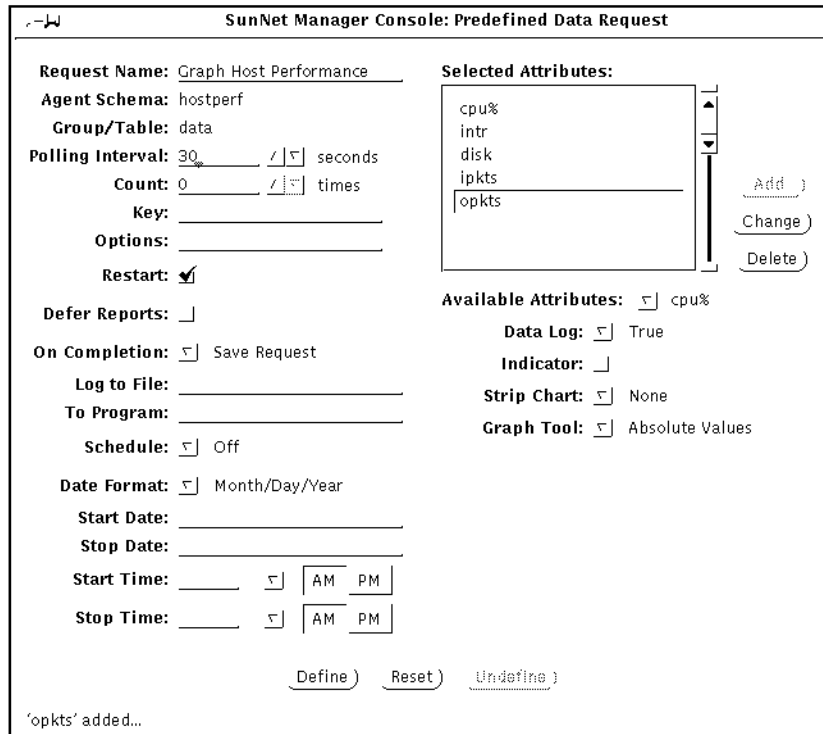


Figure 15-6 Sample Predefined Data Request Template

15.2.4.1 Reporting Characteristics

The fields on the left side of the Predefined Data/Event Request window specify reporting characteristics. Each of the fields is described below.

Request Name:

The name you are assigning to the predefined request. You *must* specify a name for this request. Request Names are limited to a maximum of 64 characters. If you selected a predefined data or event request, the Request Name field will contain the name of the predefined data or event request you selected.

Case Sensitivity

The predefined data and event record names are case insensitive. Since the predefined data and event records reside in the runtime database with all of the elements, it is *strongly* recommended that the names assigned to newly created predefined requests consist of at least two words, as is the case with the predefined data and event requests provided with this product. Otherwise, a conflict may arise between the name of an element and the name of the predefined data and/or event request record being created, as they must be unique. When an attempt to create a predefined data or event request with a duplicate name of an element or an existing predefined request is made, an error message will be placed in the error reports log and the predefined request will not be created.

Agent Schema:

The name of the agent schema with which the predefined data request is associated.

Group/Table:

The name of the group or table associated with the agent schema with which the predefined data request is associated.

Polling Interval:

Specifies the interval (in seconds) for the agent on the target system to send reports. An Interval value of 0 indicates that the agent should use its default interval. See the specific agent's `man` page for its default interval value.

Count:

Specifies the number of times for the agent on the target system to send reports. A count of 0 specifies that the agent is to send reports until the request is stopped or killed. If a count is specified, and a Stop Date/Time is/are specified, the request stops if the specified number of reports are generated before the Stop Date/Time are reached. If the Stop Date/Time is/are reached before the specified number of reports are generated, the request stops at the specified time.

Key:

Identifies a particular row in a table (for example, `ie0` or `le0` in an Ethernet interface table). The use of a key is agent-specific and is indicated in the bottom left corner of the Data Request Properties window. See the `man` page for the particular agent for information regarding keys. If no key is specified, the entire table is returned.

Options:

Specifies any options (such as arguments) that the agent expects. To specify Options, type in an option string. For example, with SNMP requests you can specify an SNMP read-community name. The information that you can specify in the Options field is agent-specific—not all agents accept options. See the `man` page for the particular agent for information regarding options.

Restart:

Specifies whether the agent should attempt to restart the request if the system on which the agent is running reboots, the agent terminates unexpectedly, or the Console itself is restarted. If Restart is off (no check mark appears in the accompanying box), the request is discarded if the agent fails or if you quit the Console. If Restart is on (a check mark appears in the accompanying box), the request is restarted. The default setting of this field is determined by the Restart Request upon Agent Failure setting in the Console Properties Requests category. (Refer to Chapter 17, “Props Menu,” under “Requests” for more information.) Click SELECT on the box next to this field to toggle the check mark on or off.

Defer Reports:

Specifies whether or not the agent should cache reports and send them to the Console only when asked to. If Defer Reports is on (a check mark appears in the accompanying box), the agent is directed to collect the statistics, but send them to the Console only when directed. Defer Reports tells the agent to cache the last 32 reports. If 32 reports are cached, or if the agent runs out of memory, the oldest report is deleted when a new report is cached. You should not specify Defer Reports for agents that return a large amount of data (for example, a routing table).

In many cases an agent collects information useful for debugging problems. This information might not be of interest in your normal daily operation. Furthermore, if a request were started before the error occurred, data reports will continually stream back to the Console causing unnecessary network traffic and increased CPU load. If, on the other hand, the request was started after the error happened, the debugging information would not have been collected. By setting Deferred Sending on, reports are held in the agent’s system until you need to ask for them.

The data request must be active for Defer Reports to work properly; that is, issue the request with a long Interval and high Count. Refer to Chapter 14, “Console,” under “Viewing and Modifying Requests” for information on how to obtain deferred reports.

On Completion:

Specifies whether the request should be deleted or saved upon completion. To specify that the request be saved, drag MENU over the On Completion abbreviated menu button—release MENU over Save Request. If you choose to save the request, the request glyph is dimmed after the request has been completed but remains in the view of the target element. You can examine, modify, or restart the saved request.

If you select Delete Request On Completion, then stopping the request is the same as killing it. (Refer to Chapter 14, “Console,” under “Viewing and Modifying Requests” for more information.)

Log to File:

Specifies the file name where reports are to be stored. If you specify a file name with no path, the SunNet Manager Console uses the current directory. If you fully qualify the file name and the left-most characters scroll left, they are not lost.

Note – Starting with version 2.3, there is a new flag called `get-requested-attribute-only`. By default it is set to true. Only the requested attributes will be retrieved. When this flag is set to false, all attributes are retrieved.

To Program:

A shell command line specifying a program or shell script to be run. Include the directory path, if necessary. Data reports are passed to the standard input of the program or shell script. Every active program to which data reports are sent keeps a file descriptor open. The default maximum number of file descriptors allowed for a single process is 64. You can increase this number up to 256 by using the `limit` command in the C-shell.

Schedule:

Set this field to “On” in order to activate the scheduling options below.

Date Format:

Select one of the four formats in which you will enter the start and stop dates.

Start Date:

When you enter a date here, and click the Start button, the request will start on the specified date.

Stop Date:

When you enter a date here, the request will stop on the specified date.

Start Time:

When you enter a time here, and click the Start button, the request will start at the specified time.

Stop Time:

When you enter a time here, the request will stop at the specified time.

15.2.4.2 *Selecting Attributes*

The fields in the right portion of the Predefined Data Request window are used to specify attributes.

Selected Attributes:

A scrolling list that contains those attributes for which you are requesting data. You choose an attribute and how you want the attribute data displayed with the fields beneath the Available Attribute values scrolling list (described below). When you click SELECT on the Add button at the right of the window, the attribute is added to the Selected Attributes values scrolling list.

Following is a description of the Available Attributes and attribute display fields:

Available Attributes:

Specifies the attribute you want the agent to return data on. When the Predefined Data Request window is first displayed, the Available Attributes field displays the name of the first attribute in the group/table from the agent schema file. To specify a particular Available Attribute, press MENU over the Available Attributes abbreviated menu button and release MENU on the desired entry. When an attribute is selected, a description is displayed in the lower-right footer.

Note – If no attribute is specified, *all* attributes for the selected group or table are written to the Data Reports log. For tables, the values of the keys are also returned. However, if you specify a particular table attribute, the key values are not returned.

Data Log:

Specifies whether or not the attribute data should be written in the Data Reports Log. To specify that the data *not* be written in the Data Reports log, drag MENU over the Data Log abbreviated menu button—release MENU over False.

Indicator:

Specifies whether or not the attribute data should be displayed in an Indicator. An Indicator shows the last reported value for a particular attribute. Refer to Chapter 16, “View Reports,” under “Indicators” for more information.

To specify the Indicator display option, click SELECT on the Indicator box.

Strip Chart:

Specifies whether or not the attribute data should be displayed in a Strip Chart. A Strip Chart contains an auto-scaled strip chart of values for an attribute. Refer to the section on “Strip Charts” for more information.

To specify the Strip Chart display option, drag MENU over the Strip Chart abbreviated menu button—release MENU over either Absolute Values (chart the received values) or Delta Values (chart the differences between received values).

Graph Tool:

Specifies whether or not the attribute data should be displayed by the Grapher. The Graph Tool, also known as the Results Grapher, is an application that allows the display of data received by the Console. Only data of types integer, float, counter, gauge, timestamp, or UNIX time can be plotted. Refer to Chapter 21, “Results Grapher,” for more information about the Graph Tool. To specify the Graph Tool display option, drag MENU over the Graph Tool abbreviated menu button—release MENU over either Absolute Values (graph the received values) or Delta Values (graph the differences between received values).

15.2.4.3 Adding Attributes

The buttons to the right of the Selected Attributes field in the Predefined Data Request window allow you to add, change, and delete attribute selections in the Selected Attributes values scrolling list. The buttons are described below:

- Add tells the request manager to add the Available Attribute selected to the Selected Attributes scrolling list.
- Change tells the request manager to apply the changes made to the attribute setting(s).
- Delete tells the request manager to remove the Available Attribute selected from the Selected Attributes scrolling list.

An entry in the scrolling list may be selected at any time by pointing the mouse pointer at the entry and clicking the SELECT button. This updates the window to the settings for the selected attribute. An entry may be deleted by selecting it and clicking on the Delete button.

15.2.5 Define and Reset Data Record Fields

Use the buttons at the bottom of the Predefined Data Request window to define and reset all the data record fields for the predefined data request, as well as to delete the record from the runtime database. The buttons are described below:

- Define creates or updates the predefined data request runtime database record to the values specified in the Predefined Data Request Window.
- Reset returns the request specifications back to the originally applied selections.
- Undefine deletes the selected predefined data request from the runtime database record.

15.2.6 Predefined Event Request Management

When you specify a predefined Event Request, the window in Figure 15-7 appears after you select the Request Type, Request Name, Agent Schema, and Group Name in the Request Builder window.

SunNet Manager Console: Predefined Event Request	
Request Name: hostmem2.eventreq	Selected Attributes:
Agent Schema: hostmem2	
Group/Table: streams	
Polling Interval: 600 second	Add
Count: 0 times	Change
Key:	Delete
Options:	
Restart: <input checked="" type="checkbox"/>	Available Attributes: strused
Send Once:	Relation1: Threshold Not Set
Defer Reports:	Threshold1:
On Completion: Save Request	Relation2: Threshold Not Set
Schedule: Off	Threshold2:
Date Format: Month/Day/Year	Priority: Low
Start Date:	Glyph Effect: Blink Glyph
Stop Date:	Audio Effect: None
Start Time: AM PM	Audio File:
Stop Time: AM PM	Mail To:
	To Program:
	Stop Request:
	Start Request:
	Define Reset Undefine

Figure 15-7 Sample Predefined Event Request Template

Descriptions of the reporting characteristics fields are provided below:

Request Name:

The optional name you wish to assign to the request. This allows you to assign a recognizable name to a request. The request manager assigns a request name, if one is not specified, using the following format:

`<agent>.<group>.<number>` (for example, `hostperf.data.0`)

Agent Schema:

The file containing the attribute group/table.

Group/Table:

the set of attribute values for which you are requesting data.

Polling Interval:

Specifies the interval (in seconds) for the agent on the target system to send reports. An Interval value of 0 indicates that the agent should use its default interval. See the agent's `man` page for its default interval.

Count:

Specifies the number of times for the agent on the target system is to send reports. A count of 0 specifies that the agent is to send reports until the request is stopped or killed. If a count is specified, and a Stop Date/Time is/are specified, the request stops if the specified number of reports are generated before the Stop Date/Time are reached. If the Stop Date/Time is/are reached before the specified number of reports are generated, the request stops at the specified time.

Key:

Identifies a particular row in a table (for example, `ie0` or `le0` in an Ethernet interface table). The use of a key is agent-specific and is indicated in the bottom left corner of the Data Request Properties window. See the `man` page for the particular agent for information regarding keys. If no key is specified, the entire table is returned.

Options:

Specifies any options (such as arguments) that the agent expects. To specify Options, type in an option string. For example, with SNMP requests you can specify an SNMP read-community name. The information that you can specify in the Options field is agent-specific—not all agents accept options. See the `man` page for the particular agent for information regarding options.

Restart:

Specifies whether the agent should attempt to restart the request if the system on which the agent is running reboots, the agent terminates unexpectedly, or the Console itself is restarted. If Restart is off (no check mark appears in the accompanying box), the request is discarded if the agent fails or if you quit the Console. If Restart is on (a check mark appears in the accompanying box), the request is restarted. The default setting of this field is determined by the Restart Request upon Agent Failure setting in the Console Properties Requests category. Click SELECT on the box next to this field to toggle the check mark on or off.

Send Once

If Send Once has been selected (a check mark appears in the accompanying box), the request is terminated after the Console receives the first event report. Click SELECT on the Send Once box to toggle the check mark on or off.

Defer Reports:

Specifies whether or not the agent should cache reports and send them to the Console only when asked to. If Defer Reports is on (a check mark appears in the accompanying box), the agent is directed to collect the statistics, but send them to the Console only when directed. Defer Reports tells the agent to cache the last 32 reports. If 32 reports are cached, or if the agent runs out of memory, the oldest report is deleted when a new report is cached. You should not specify Defer Reports for agents that return a large amount of data (for example, a routing table).

In many cases an agent collects information useful for debugging problems. This information might not be of interest in your normal daily operation. Furthermore, if a request were started before the error occurred, data reports will continually stream back to the Console causing unnecessary network traffic and increased CPU load. If, on the other hand, the request was started after the error happened, the debugging information would not have been collected. By setting Deferred Sending on, reports are held in the agent's system until you need to ask for them.

The data request must be active for Defer Reports to work properly; that is, issue the request with a long Interval and high Count. Refer to the “Viewing and Modifying Requests” section for more information on how to obtain deferred reports.

On Completion:

Specifies whether the request should be deleted or saved upon completion. To specify that the request be saved, drag MENU over the On Completion abbreviated menu button—release MENU over Save Request. If you choose to save the request, the request glyph is dimmed after the request has been completed but remains in the view of the target element. You can examine, modify, or restart the saved request.

If you choose Delete Request on Completion, stopping the request is the same as killing it. (Refer to the “Viewing and Modifying Requests” section for more information.)

Schedule:

Set this field to “On” in order to activate the scheduling options below.

Date Format:

Select one of the four formats in which you will enter the start and stop dates.

Start Date:

When you enter a date here, and click the Start button, the request will start on the specified date.

Stop Date:

When you enter a date here, the request will stop on the specified date.

Start Time:

When you enter a time here, and click the Start button, the request will start at the specified time.

Stop Time:

When you enter a time here, the request will stop at the specified time.

Important Information About Polling Interval and Count Fields

For predefined event requests, these fields specify when the agent is to send a report. A report is forwarded to the Console system only if it has been determined that an event has occurred. For agents that are shipped with the product, the value reported for an attribute is the value noted at the reporting interval. Thus, it is possible for event conditions to occur *between* reporting intervals and not cause an event report.

If a count is specified, and a Stop Date/Time is/are specified, the request stops if the specified number of reports are generated before the Stop Date/Time are reached. If the Stop Date/Time is/are reached before the specified number of reports are generated, the request stops at the specified time.

15.2.6.1 Selecting Attributes

Use the fields in the right portion of the Predefine Event Request properties window to specify attributes.

Selected Attributes:

A scrolling list that contains those attributes for which you are requesting event information. You specify attributes, event thresholds, and how you want the event signalled with the fields under the Selected Attributes values scrolling list. When you click SELECT on the Add button to the right of the Selected Attributes window, the attribute is added to the Selected Attributes values scrolling list.

Following is a description of the available attributes, event threshold, signal options, receiver, and priority fields:

Available Attributes:

Specifies the attributes that you want to use to specify an event. When the window is first displayed, the Available Attributes field displays the name of the first attribute in the group/table from the agent schema file. To specify an attribute, press MENU on the Available Attributes field and release the mouse button on the desired entry. When an attribute is selected, a description is displayed in the lower-right footer of the window.

Relation1, Relation2:

Specify threshold conditions for reporting an event. You can specify up to two threshold relational operators; the event is reported if *either* of the two threshold conditions are met. To set the threshold relational operator(s) for

Relation1 or Relation2, drag MENU over the Relation abbreviated menu button to display the following list of relational operators in a pop-up menu—release MENU over the desired relational operator.

Threshold Not Set means you have not set a threshold for this attribute. This is the default choice.

Equal To tells the agent to report an event if the attribute value equals “Threshold n .”

Not Equal To tells the agent to report an event if the attribute value does not equal “Threshold n .”

Less Than tells the agent to report an event if the attribute value is less than “Threshold n .”

Less Than Or Equal To tells the agent to report an event if the attribute value is less than or equal to “Threshold n .”

Greater Than tells the agent to report an event if the attribute value is greater than “Threshold n .”

Greater Than Or Equal To tells the agent to report an event if the attribute value is greater than or equal to “Threshold n .”

Changed tells the agent to report an event if the attribute value differs from the last sampled value.

Increased By tells the agent to report an event if the attribute value has increased by “Threshold n ” compared to the last sampled value.

Decreased By tells the agent to report an event if the attribute value has decreased by “Threshold n ” compared to the last sampled value.

Increased By More Than tells the agent to report an event if the attribute value has increased by more than “Threshold n ” compared to the last sampled value.

Increased By Less Than tells the agent to report an event if the attribute value has increased by less than “Threshold n ” compared to the last sampled value.

Decreased By More Than tells the agent to report an event if the attribute value has decreased by more than “Threshold n ” compared to the last sampled value.

Decreased By Less Than tells the agent to report an event if the attribute

value has decreased by less than “Threshold n ” compared to the last sampled value.

Note – For the Changed, Increased By, Decreased By, Increased By More Than, Increased By Less Than, Decreased By More Than, and Decreased By Less Than operators, the Count field of the report characteristics must not be ‘1.’ A Key must be supplied if a table of attributes is selected.

Threshold1, Threshold2:

Specify the threshold value if the selected operator is other than Threshold Not Set, or Changed.

Priority:

Specifies the priority of the request. Click SELECT on the desired priority button (Low, Medium, or High) for the request.

Glyph Effect:

Specifies a visible indicator when the specified event is reported. Press MENU on the Glyph Effect abbreviated menu button to display a signal options pop-up menu. Release MENU on the signal option you want. You can choose only one visual signal option. In order to have the “decay” feature in effect, you must select the Priority by Color option.

The following options are available:

Blink Glyph

Blinks the glyph of the element for which an event has been reported.

Dim Glyph

Dims the glyph of the element for which an event has been reported.

Priority by Color

Causes the glyph to change color, based on the priority of the reported event. The defaults are:

- Red = high priority
- Orange = medium priority
- Yellow = low priority
- Blue = decay

Starting with version 2.3, you can customize colors, including the “decay” color. See Chapter 4, “Requesting Data,” or Chapter 5, “Specifying Event Requests,” for procedures to customize Color by Priority.

If multiple reports are received for a system, the highest priority is used. In association with the Priority by Color option is the “decay” feature. This feature pertains to events and traps that have reached a threshold and then fallen below it, or that have had an event or trap stop being reported. In response to these conditions, the glyph for the affected element turns blue, which is the default color, or to the color you have customized. Once a glyph has decayed to blue, the event must be acknowledged in order for the glyph to return to its original color. To return a glyph to its original color, you can either:

- Select the Console’s View►Event Summary option
 - Highlight the name of the glyph
 - Press the Drop from List button.
- Or
- From the Glyph Menu for the specific element, pull right over the Glyph State►Normal option.

Pending State

Starting with version 2.3, you can place an object in pending state. In this state, the color of the glyph is dimmed, and all outstanding events are cleared. New events/traps do not change the color of the glyph nor do they propagate the effect of the trap/event to the parent object. A parent object in pending state is not affected by a change in state on any children.

Audio Effect:

Specifies an audible indicator when the specified event is reported. Press MENU on the Audio Effect abbreviated menu button to display a signal options pop-up menu—drag and release MENU on the desired signal option. Only one audio signal option can be chosen. The following options are available:

Ring Bell sounds an audible bell on the Console workstation.

Play Audio File causes the file specified in the Audio File field to be played.

Audio File:

Defines the audio file to be played when the specified event is reported. Type in the file name of the audio file. The option `Play Audio File` must be specified in the `Audio Effect` field. You can use the volume function of the `Audio Tool` in `OpenWindows 3.2` to control the sound level.

Note – The Console must be running (not merely displaying) on a machine with an audio port. If you are running the Console on a server but displaying the Console windows on a local workstation, the Console will attempt to play the audio file on the server.

Mail To:

Defines one or more mail recipients to send the event report to when the specified event is reported. Type in a list of mail recipients. If there is more than one recipient, use a space between each entry.

To Program:

A shell command line specifying a program or shell script to be run. Include the directory path, if necessary. A new copy of the program or shell script is forked for each event report. The event report is passed to the standard input of the program or shell script. Every active program that receives an event report keeps a file descriptor open. The default maximum number of file descriptors allowed for a single process is 64. You can increase this number up to 256 by using the `limit` command in the C-shell.

Stop Request:

This field is new starting with version 2.3. If this field is checked, the request will be launched one time. If this field is checked, it will override any number specified in the `Count` field.

Start Request:

This field is new starting with version 2.3. This field defines a child request that is to be launched when the parent request in the `Request Name` field is launched. When this sequence is launched, the child request becomes a parent request enabling you to specify another child request, and so forth. In this way, you can launch as many requests as you choose.

15.2.6.2 Adding Attributes

The buttons to the right of the Selected Attributes field in the Predefined Event Request window allow you to add, change, and delete attribute selections in the Selected Attributes values scrolling list. The buttons are described below:

- Add tells the request manager to add the Available Attribute selected to the Selected Attributes scrolling list.
- Change tells the request manager to apply the changes made to the attribute setting(s).
- Delete tells the request manager to remove the selected attribute from the Selected Attributes scrolling list.

An entry in the scrolling list may be selected at any time by pointing the mouse pointer at the entry and clicking the SELECT button. This updates the window to the settings for the selected attribute. An entry may be deleted by selecting it and clicking on the Delete button.

15.2.6.3 Define and Reset Event Record Fields

The buttons at the bottom of the Predefined Event Request window allow you to define and reset all of the event record fields for the predefined event request, as well as delete the record from the runtime database. The buttons are described below:

- Define creates or updates the predefined event request runtime database record to the values specified in the Predefined Event Request window.
- Reset returns the request specifications back to the originally applied selections.
- Undefine deletes the selected predefined event request from the runtime database record.

15.3 Requests Summary

To get a summary of the data and event requests, click SELECT on the Requests Summary button. The window shown in Figure 15-8 on page 15-41 appears. For a description of the fields in the Requests Summary Window, refer to Chapter 16, “View Reports.”

To display the Requests Summary window, press MENU over the Console Requests►Requests Summary option and release.

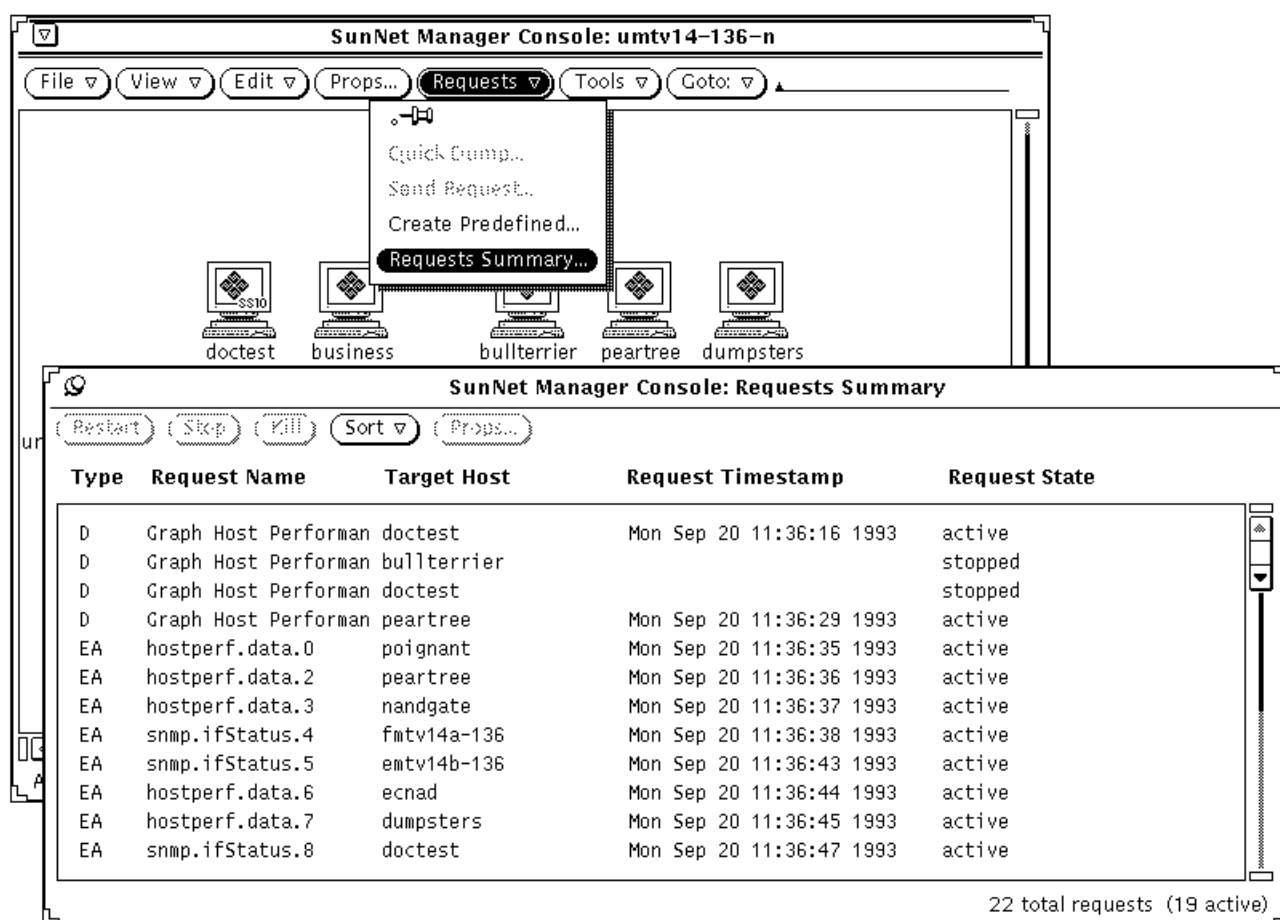


Figure 15-8 Requests Menu—Requests Summary Window

The following information is displayed in the Requests Summary window:

- Type defines the request as one of the following:
 - D: data request
 - E: event request

- EA: event request launched by the Automatic Node Management feature of the Console
- Request Name lists the name you have assigned to the request or, if you have not assigned a name, a name of the format `<agent>.<group>.<number>`, for example, `ping.reach.4`.
- Target Host is the name of the target element for the request.
- Request Timestamp is the date that the request was started or restarted. Dates are only displayed for requests that have been started at least once.
- Request State is one of the following:
 - Active
 - Stopped
 - Awaiting stop
 - Stopping
 - Awaiting activation
 - Being activated
 - Scheduled for later

The footer of the window indicates the total number of requests (whether active or stopped) and the number of active requests.

Using the buttons at the top of the window, you can restart, stop, sort, or kill requests. The operation applies to all requests that you have selected (highlighted). To select requests, press MENU in the scrolling list area of the window to bring up the Select menu. The Select menu is shown in Figure 15-9.

If the On Completion field in the properties of a request is set to Delete Request (rather than the default Save Request), then stopping that request is the same as killing the request. A description of the fields in requests can be found in Section 15.1.0.1, "Data Request Template."

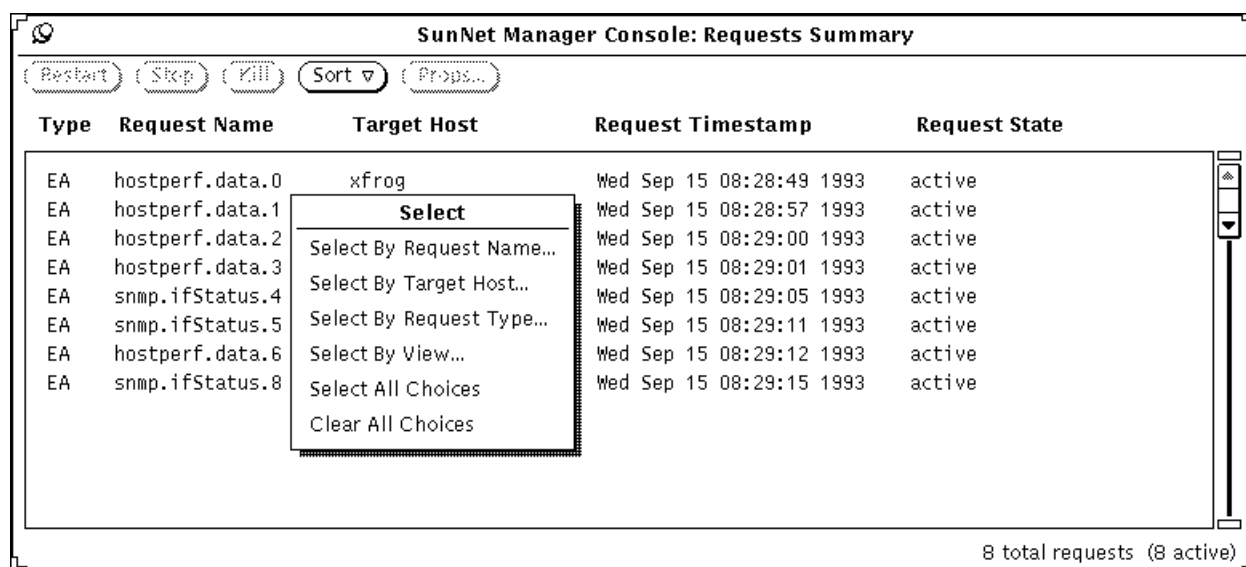


Figure 15-9 Selecting Requests

If you choose Select By Request Name, Select By Target Host, Select by Request Type, or Select By View, a pop-up window is displayed where you can specify the request name, target element, request type, or view name. Wild card characters (* and ?) may be used when specifying the request name, target element, or request type. View name cannot be specified with wild card characters.

Note – If an element is selected (highlighted) in a Console view, the element name appears in the target element or view name pop-up window.

Use the Sort button menu to sort the displayed requests by request name, target host, request timestamp, request type, or request state. Sorting by timestamp displays the requests from earliest to latest start times. Sorting by request state displays requests according to the following order of request states:

1. Stopped
2. Awaiting activation
3. Being activated
4. Active
5. Awaiting stop
6. Stopping
7. Scheduled for later

All other sorting is done in alphabetical order.

Use the Props button to display the Request Properties window. You can then modify any of the properties of the request. If more than one request is selected when you click SELECT on the Props button, only the Properties window for the first selected request is displayed.

15.4 Request Glyph Popup Menu

Requests are represented by glyphs in the subview of the target element. To display the request glyph view, you can either double-click SELECT on the target element or choose the element Glyph►Show Subview menu option.

Note – Held requests or requests that are completed and saved are displayed as dimmed.

There is a pop-up menu associated with each request glyph. To display the request glyph's menu, move the mouse pointer over a the request glyph and press MENU. This is shown in Figure 15-10.

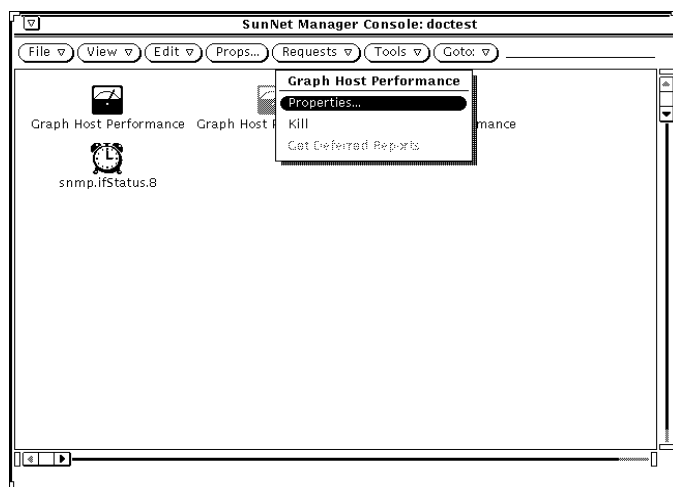


Figure 15-10 Request Glyph Menu

Following is a description of the options in the Request Glyph Menu:

- Properties shows the properties of a particular request. Figure 15-11 is an example.

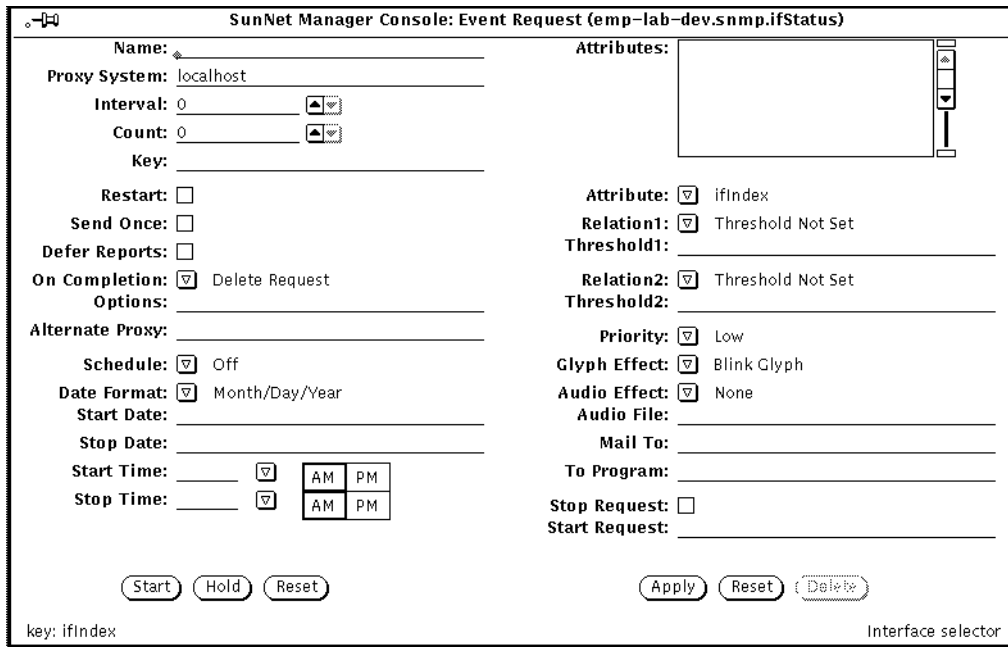


Figure 15-11 Sample Request Properties Window

You can modify the request in several ways. For example, you can choose a different attribute, change the report frequency, stop an active request, or start a request that was placed on hold. When you modify a request, you are actually changing the specifications for that request, not launching a new one. The request manager stops the original request and restarts it with the new specifications.

Clicking the Stop button stops the request and saves it only if you have previously specified that the request be saved upon completion. Otherwise, the Stop button effectively kills the request. Use the Restart button to launch held or saved requests. If the request has already been started, Restart restarts the modified request with the new attribute specification and report characteristics.

- Kill allows you to kill a request that you no longer wish to run. Kill deletes a request even if you have previously set the Upon Completion request properties field to Save Request.

Note – Quitting the Console causes any active requests to be discarded unless the Restart field is on (a check mark appears in the accompanying box) or the On Completion field is set to Save Request.

- Send Deferred indicates you are now ready to receive deferred reports. The agent system now sends the deferred reports (up to 32) to the Console. Refer to the Defer Reports discussion in the "Reporting Characteristics" section of Chapter 20, "Browser," for a description of deferred reports.

View Reports

This chapter discusses the following topics:

- The View button for alarm reports
- The View button for data reports
- The View button for event reports

Use the View button to:

- Display a list of the alarm reports received.
- Display a log of the data reports received.
- Display a log of the event/trap reports received.
- Display a log of the error reports received.
- Display a list of machines that have had an event occur on them.
- Specify a background image to be displayed for the current view.
- Remove a background image from the current view.
- Find a particular element.
- Look at the contents of the clipboard.

See Figure 16-1 for a sample of the View Button Menu.

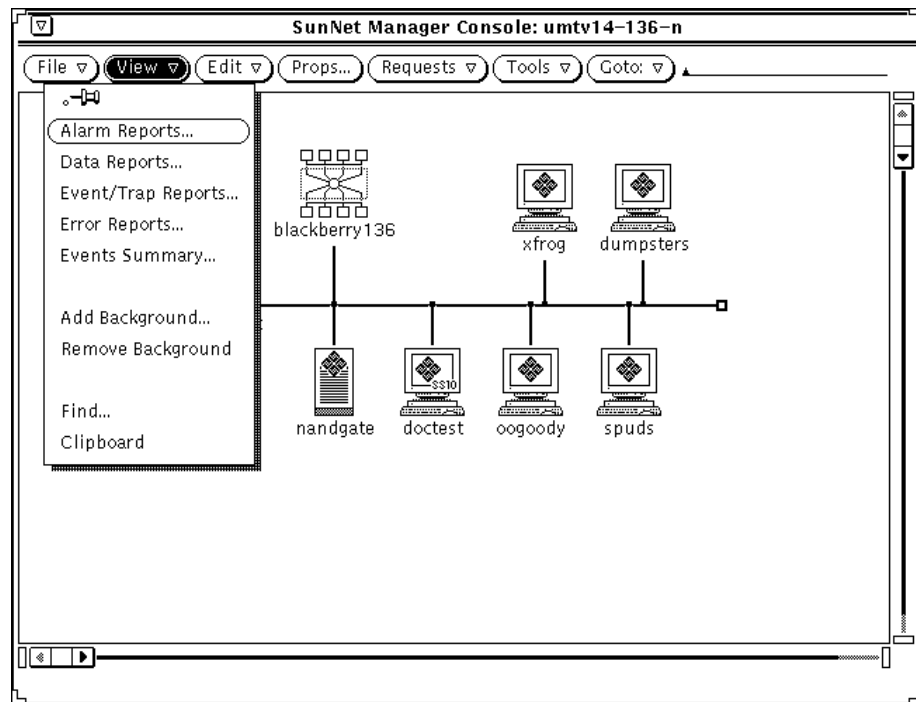


Figure 16-1 View Button Menu

16.1 Alarm Reports

Alarm Reports facilitates alarm/fault management. It displays the number and priority (high/medium/low) of events, traps, and errors to have occurred. The reports can pertain to a specific element or to an entire view.

SNM allows you to view a summary of alarm reports for all elements within a given view. Event, trap, and error reports can be summarized in Figure 16-2.

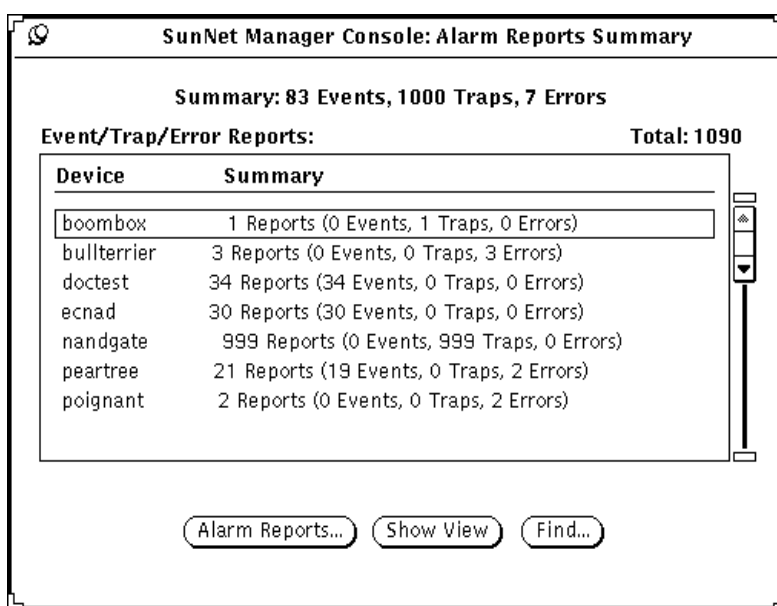


Figure 16-2 Alarm Reports Summary Window

The Alarm Reports Summary window is updated every 30 seconds. You might find it useful to leave the window pinned up while the Console is running. The Alarms Reports Summary displays reports only for the current view. If there are no reports pertaining to the current view, the Alarms Reports Summary window is blank. Upon switching to another view, the Alarms Reports Summary window is updated with the switched-to view's alarm reports summary, if such reports exist.

Note – If both the Console Alarm Report Summary window and a device specific Alarm Summary window are being displayed simultaneously and reports continue to arrive for the device, the number of reports received totals for the two windows will differ until the Console Alarm Reports Summary window is next updated (every 30 seconds.)

The Alarm Reports Summary window has three buttons at the bottom, described in the following sections.

16.1.1 Alarm Reports

Allows you to view a device-specific alarm report for a device selected in the Alarm Reports Summary window's scrolling list.

A device-specific Alarm Reports window allows you to:

- Sort alarm reports
- Filter reports
- Find reports associated with a specific agent
- Save reports to a file (which can be viewed using the Results Browser)
- Print reports

These functions are available through the View, Save, and Print buttons at the bottom of the Alarm Reports window.

Sort Reports

In a device-specific Alarm Reports window, press MENU on View▶Sort By. You receive the following menu:

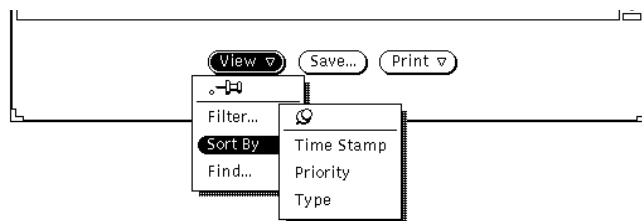


Figure 16-3 Device-specific Alarm Reports Sort Menu

These choices are mutually exclusive. Release MENU over the sorting method you want. Your selection takes effect immediately.

The sorting choices are explained as follows:

Time Stamp

The default choice. Reports are sorted in chronological order with the most recent being last on the list.

Priority

Event reports are sorted according to priority—high, medium, or low—with high priority reports listed first on the list. This option makes most sense if you are filtering out trap and error reports. If you are not filtering out reports and specify sorting by priority, event reports come first, followed by error reports, then trap reports.

Type

Reports are sorted by type, in the order: event, error, trap. Note that alarm reports that are received after you specify sorting by type are appended to the appropriate list of reports, depending on the new reports' type.

Note – When sorting alarm reports, only the reports currently displayed in the Alarm Reports window will be sorted. All new reports received during the sorting operation are appended to the end of the sorted list.

Filter Reports

Press MENU in the View button. Release MENU over Filter to receive the following window:

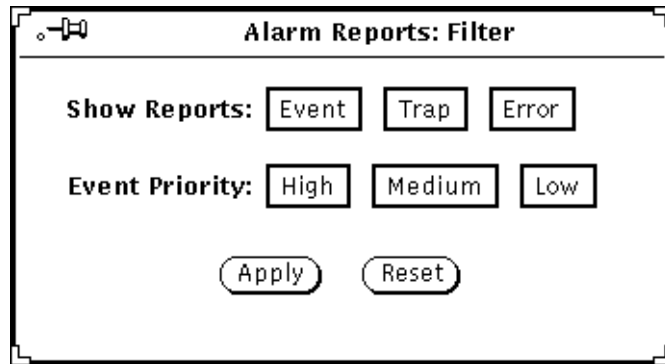


Figure 16-4 Device-specific Alarm Reports Filter Window

Your selections in the Filter window determine which types of reports are displayed in the device-specific Alarm Reports window. The default is that all alarm reports are displayed. Note that priority choices apply only to Event Reports. If you click on Apply, your choices take effect immediately. Click on Reset to restore the choices to the way they were when you brought up the Filter window.

Note – If you plan to both sort (described in “Sort Reports” on page 16-4”) and filter your device-specific alarm reports, you must sort before filtering. If you want to sort again, after filtering, you must click SELECT on Reset, then Apply, in the Filter window, sort again, then reinvoke Filter.

Find Agent-specific Reports

SNM allows you to search on an agent name to find instance(s) of event reports originating with that agent. To do this, in the device-specific Alarm Reports window, press MENU on View►Find and release MENU. In the Agent Name field, enter the agent *and* group name—not just the agent name. For example, specify `hostperf.data`, not just `hostperf`. When you click SELECT on Find, the window displays the contents of the first (oldest) report originating with the agent you specified. If there is no report originating with the agent you specify, you receive the message, “Agent name not found.”

Save Device-specific Reports

SNM allows you to save device-specific alarm reports to a file. To do this, in the device-specific Alarm Reports window, click SELECT on Save. You receive a window such as the one in Section Figure 16-5, “Save to Logfile Window.”

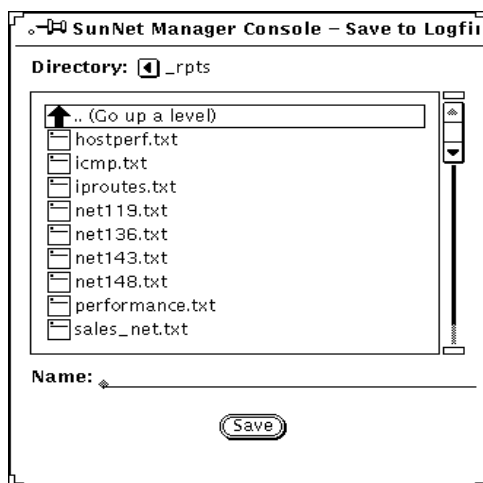


Figure 16-5 Save to Logfile Window

If the directory that is first displayed is not the directory you want, change directories by clicking SELECT on folder icons or the .. (level above) symbol. Alternatively, you can enter a path name in the Name field and click SELECT on Save. If the path you enter ends in a valid directory, you are switched to that directory.

When you reach the directory you want, enter the name of the file to which the report will be written.

Print Device-specific Reports

SNM allows you to print device-specific alarm reports. Click SELECT on the Print button on the bottom of the device-specific Alarm Reports window, or press MENU on Print►Report and release MENU. SNM sends the currently displayed alarm report to your default printer.

16.1.2 Show View

After having selected a specific device from the scrolling list, clicking SELECT on the Show View button selects the specified device and brings it into view in the Console window and tells you whether the device resides under one or multiple views. You receive a window such as the one shown in Figure 16-6.

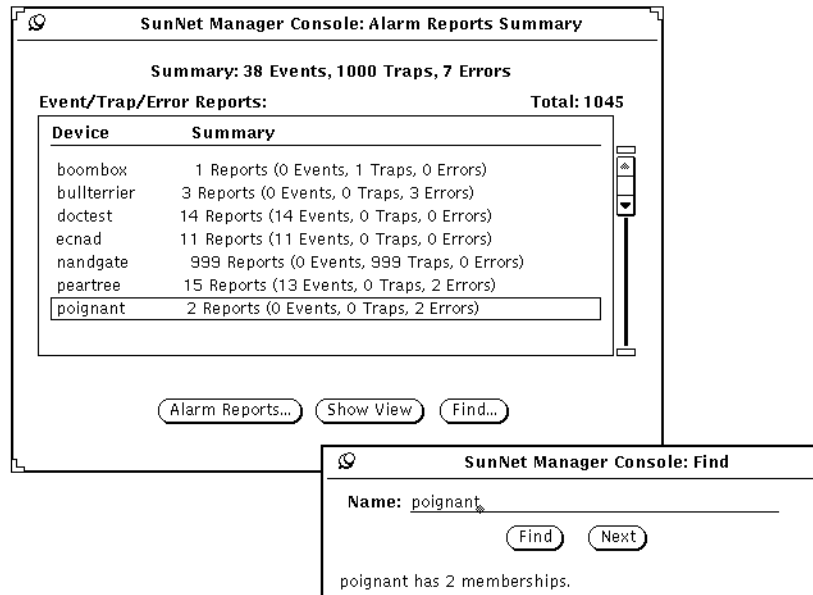


Figure 16-6 Alarm Reports Show View Window

If a device resides under multiple views, clicking the Next button allows you to switch to the next view under which this device resides.

If a device resides under multiple views, clicking the Next button allows you to switch to the next view under which this device resides.

If a device is in only one view, the Next button in the window above is grayed-out. If a device is in multiple views, click SELECT on Next to switch from the current view (as displayed in the Console window) to the next view in which the specified device appears. In this context, “next” means the next view on the list of views as displayed by the Console’s Goto menu.

16.1.3 Find

Allows you to find an element entry not visible in the Alarm Reports Summary window's scrolling list.



Figure 16-7 Alarm Reports Find Window

To find a device that is not visible in the window's scrolling list, you can scroll through the list or, more conveniently, click SELECT on the Find button at the bottom of the window

If a line for a device is present, that line is displayed and highlighted in the Alarm Reports Summary window's scrolling list. If a line for a device is not present, you receive the message "Device name not found" in the Alarm Reports Summary Find pop-up window.

This section describes several ways of viewing output from data and event requests. Additionally, you can view data reports or event/trap reports with the SunNet Manager Results Browser and Results Grapher. (See Chapter 20, "Browser," and Chapter 21, "Results Grapher," for more information.)

16.2 Data Reports

For data requests, you specify how you want the attribute data displayed (data reports log, Strip Chart, or Indicator) in the Data Request window. You view the Data Reports log through the SNM Console View menu, while Strip Charts and Indicators are displayed in the view from which the data request was launched. The following subsections discuss each of these types of displays.

16.2.1 Data Reports Log

The Data Reports log contains data reports, in ASCII, for all elements managed by the SunNet Manager Console. You display the Data Reports log by selecting the SNM Console View►Data Reports menu item.

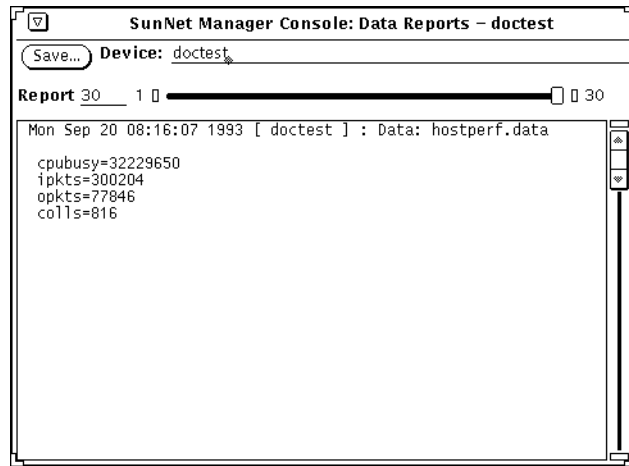


Figure 16-8 View -- Data Reports Window

Data report entries have the following format:

`<timestamp> [<element_name>] : <report type>: '<request_name>'`

Table attributes are displayed in tabular form when possible. Attributes that are not tables are displayed in the format:

`<attribute_name> = <attribute_value>`

To view data reports, drag the slider to the right or the left. To look at all data reports for all elements, leave the Device field blank. To examine data reports for a particular element instance, type in the element name after the Device prompt and press Return. If you wish to look at Data Report log entries for all the elements again, press ctrl-u to clear the Device field and press Return.

The Data Reports log contains the most recently-received data reports for all systems. The currently displayed report is indicated on the Report line, while the total number of entries is displayed at the right of the slider. If you specify an element name in the Device field, the number of entries reflects the total number of entries for that element.

Data reports are allocated a maximum number of entries in the Data Reports log. By default, the maximum number of data reports is 1000. When the allocation limit is reached for either data reports, the oldest data report or error message is deleted. You can change the maximum number of entries by modifying the Maximum Data Reports setting in the Console Properties Miscellaneous category.

Save writes the contents of the Data Reports log to disk. Selecting this button displays a pop-up window in which you can specify the file name and path of the disk file. You can use Save after specifying a Device name to save a subset of the Data Reports log to disk.

Use the scroll elevator to scroll through an entry that spans multiple panes of the Data Report window. Any number of Data Report log windows can be displayed simultaneously, allowing you to view the log entries for individual systems side-by-side.

Note – The contents of the Data Report log window are subject to configured XView limitations. This may cause data requests for very large tables to not be displayed in the Data Reports log window. If this happens, increase the value of the `text.maxDocumentSize` parameter in your `.xdefaults` file, reload the resource into the X resource manager, and restart the Console.

16.2.2 Indicators

An Indicator shows the last reported value for a particular attribute. Indicators are displayed in the format of `<label>:<value>`, where the `<label>` consists of the element name, agent name, and attribute. For example:

```
poignant.etherif.ipkts: 6892892
```

The SunNet Manager Console displays a maximum of 64 indicators in the View where the data request selection process is begun. Indicators are positioned in the View using the same algorithm applied to newly created elements.

Indicators may be edited (cut, copied, or pasted) in the same manner as other elements. For example, you can position an Indicator adjacent to the glyph for the element being monitored.

As with any other element in the SunNet Manager Console, an Indicator has properties that may be modified using the Properties option. The Indicator Properties window is similar to the Properties window for an element. The middle portion of the window for selecting agents does not apply, nor do the color slide bars in the lower portion of the window. The scrollable top portion of the window displays the Name, Label, and Data fields.

If an Indicator is associated with a request that is loaded from an ASCII file and the request's Restart field is set to True, the Indicator is displayed in the view from which the request is launched. This is the case even if you have moved the Indicator into a different view before saving the ASCII file.

16.2.3 Strip Charts

A Strip Chart contains an auto-scaled strip chart of values for a single attribute reported from a particular element.

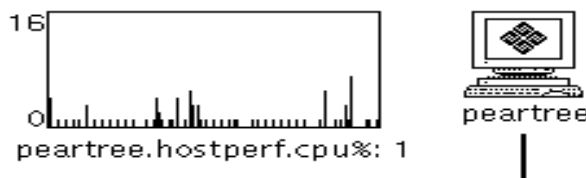


Figure 16-9 Strip Chart

The SunNet Manager Console displays up to 128 Strip Charts in the view where the data request selection process is begun. Strip Charts are positioned in the view using the same positioning algorithm applied to newly created elements.

Strip Charts may be edited (cut, copied, pasted) in the same manner as other elements. For example, all Strip Charts can be collected in a separate view.

If a Strip Chart is associated with a request that is loaded from an ASCII file and the request's Restart field is set to True, the Strip Chart is displayed in the view from which the request is launched. This is the case even if you have moved the Strip Chart into a different view before saving the ASCII file.

As with any other SunNet Manager Console element, the Strip Chart has properties that can be modified. Click MENU on the appropriate strip chart, and click SELECT on the Properties option. The window shown in the following figure will be displayed.

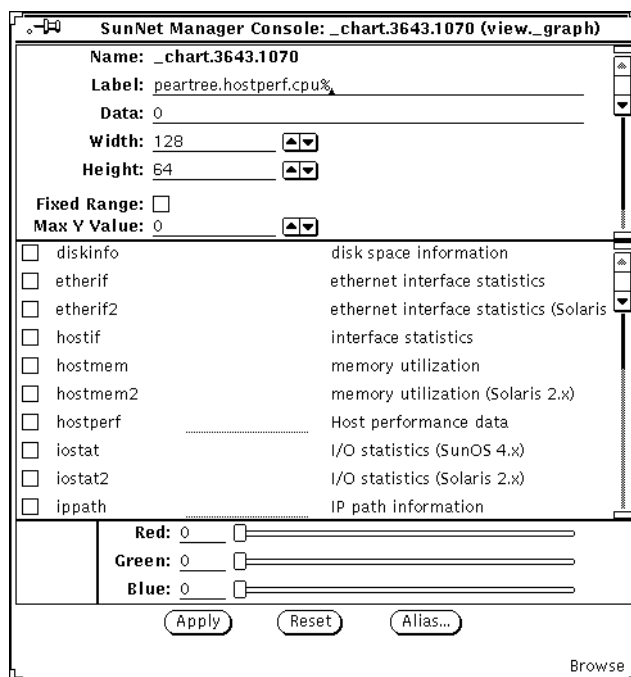


Figure 16-10 Strip Chart Properties

Note that the Strip Chart Properties window is similar to the Properties window for an element. However, the middle portion of the window for selecting agents does not apply and the strip chart color cannot be changed. The scrollable top portion of the window displays the following fields:

Name:

is an internal element identifier and may not be changed.

Label:

sets the label displayed underneath the chart.

Data:

is the latest attribute value displayed in the chart and may not be changed.

Width:

is the displayed width of the chart in pixels.

Height:

is the displayed height of the chart in pixels.

Fixed Range:

determines whether the chart is auto-ranging (no check mark in box) or fixed-range (check mark in box).

Max Y Value:

is the maximum value to use in a fixed range chart.

Min Y Value:

is the minimum value to use in a fixed range chart.

16.2.4 Graph Tool

The Graph Tool, also known as the Results Grapher, is an application that allows the display of real-time data. See Chapter 21, “Results Grapher,” for more information about displaying information with the Grapher.

16.3 Event/Trap Reports

Event/Trap Reports displays event reports as well as trap reports. The information in the reports consists of timestamp, element name, request name, and attribute name/value pairs. The reports can pertain to a specific element or to an entire view.

The Event/Trap Reports log displays reported values for selected attributes, as well as asynchronous (trap) reports. You display the Event/Trap Reports log by selecting the SNM Console View ► Event/Trap Reports menu item.

16.3.1 Event/Trap Reports Log

The Event/Trap Reports log contains event and trap reports, in ASCII, for all elements managed by the SunNet Manager Console. Event/trap reports entries are time-stamped and show attribute values for the entire agent group. The attribute and value that caused the event are flagged with the relational operator and threshold value shown to the right in parentheses. See Figure 16-11.

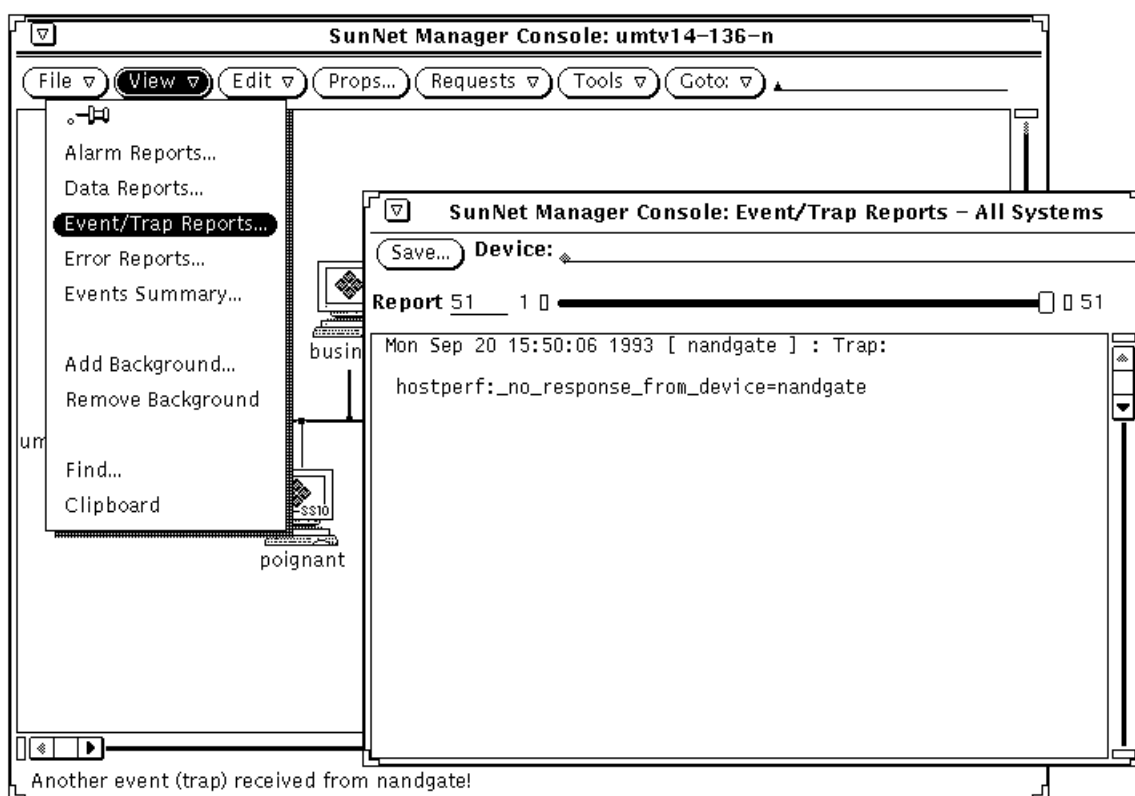


Figure 16-11 View — Event/Trap Reports

To view event/trap reports, drag the slider to the right or the left. To look at all Event/Trap log entries for all elements, leave the Device field blank. To examine event/trap reports for a particular element instance, type in the

element name after the Device prompt and press Return. If you wish to look at Event/Trap log entries for all the elements again, press ctrl-u to clear the Device field and press Return.

The Event/Trap log contains the most recently received event reports and trap reports for all systems. The currently displayed report is indicated on the Report line, while the total number of entries is displayed at the right of the slider. If you specify an element name in the Device field, the number of entries reflects the total number of entries for that element.

Event reports and trap reports are each allocated a maximum number of entries in the Event/Trap log. By default, the maximum number of reports for either events or traps is 1000. When the allocation limit is reached for either event reports or trap reports, the oldest event report or trap report is deleted. You can change the maximum number of entries by modifying the Maximum Event Reports or Maximum Trap Reports settings in the Console Properties Miscellaneous category.

Save writes the contents of the Event/Trap Reports log to disk. Selecting this button displays a pop-up window in which you can specify the file name and path of the disk file. You can use Save after specifying a Device name to save a subset of the Event/Trap Reports log to disk.

Use the scroll elevator to scroll through the entry that spans multiple panes of the Event/Trap log window. Any number of Event/Trap log windows can be displayed simultaneously, allowing you to view log entries from individual systems side-by-side.

16.4 Error Reports

Error Reports displays messages that are sent when a request is not successfully launched and when other miscellaneous errors occur. The reports can pertain to a specific element or to an entire view.

16.4.1 Errors Log

The Errors log contains a list of machines, in ASCII, that have had at least one error occur on them, for all elements managed by the SunNet Manager Console. You display the Error Reports log by selecting the SNM Console View►Error Reports menu item. You are then able to use the View►Event/Trap Reports

menu item to specify a specific device name in the Event/Trap Reports window to see the event(s) that have occurred on the machine(s) of your choice.

16.5 Events Summary

Events Summary displays a list of machines that have had at least one event or trap occur on them. You can use the Event/Trap Reports option to see the event(s) and trap(s) that have occurred for a specific device.

16.5.1 Events Summary Log

The Events Summary log contains a list of machines, in ASCII, that have had at least one event occur on them, for all elements managed by the SunNet Manager Console. You display the Events Summary log by selecting the SNM Console View►Events Summary menu item. You are then able to use the View►Event/Trap Reports menu item to specify a specific device name in the Event/Trap Reports window to see the event(s) that have occurred on the machine(s) of your choice. If you press the MENU button over the Events Summary, the Drop All popup menu appears, as shown in Figure 16-12. Selecting Drop All clears all outstanding events.

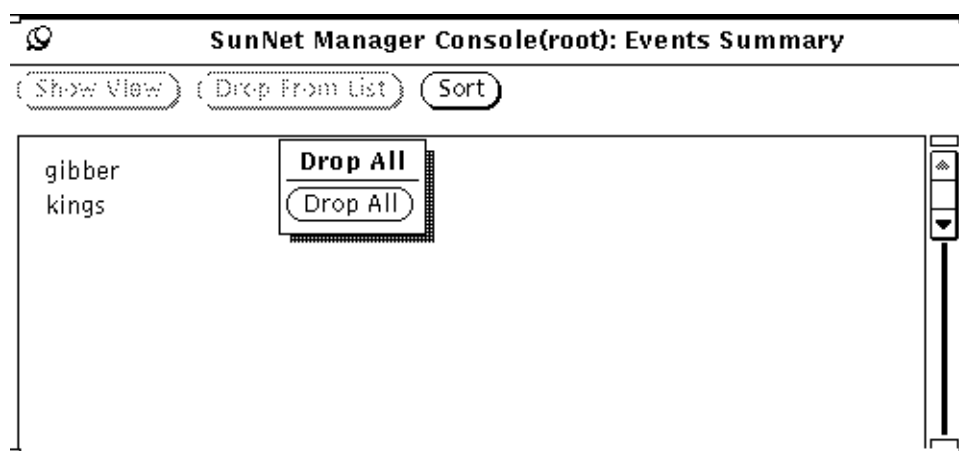


Figure 16-12 Drop All Option Popup Menu

You can select multiple devices and drop all events against them.

16.6 *Add Background*

Add Background allows you to specify the background image to be used for the current view. The background image is a Raster file.

16.7 *Remove Background*

Remove Background allows you to remove the background image from the current view.

16.8 *Find*

Find displays a window which allows you to specify an element to find (see Figure 16-13). It then displays the view containing the glyph for the element specified. The message at the bottom of the Find window tells you the number of views that the particular glyph appears in. If the glyph appears in only one view, the Next button is dimmed. Otherwise, the Console displays the next view in which the glyph appears when Next is selected.

Note – Element names are *not* case-sensitive.

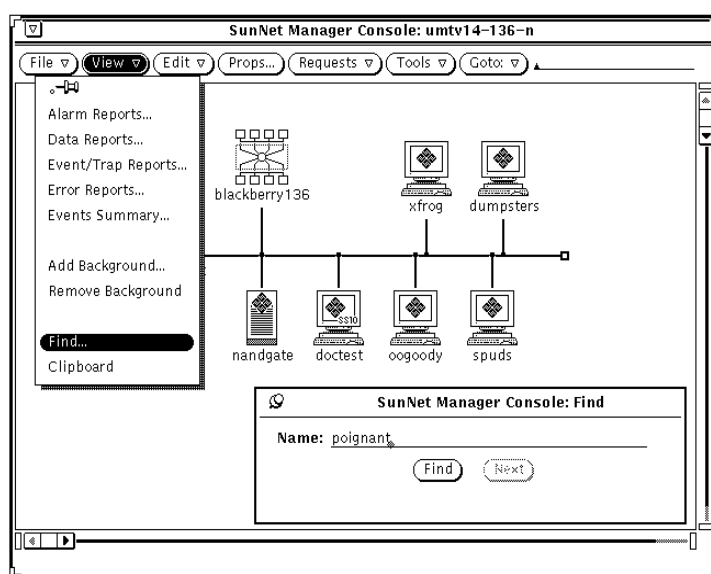


Figure 16-13 View—Find Window

16.9 Clipboard

Clipboard displays the contents of the clipboard. For example, glyphs for elements and requests edited via a Cut or Copy operation are displayed. See “Edit Button” in the following subsection for a discussion of the Cut and Copy operations.

This section describes several ways of viewing output from data and event requests. Additionally, you can view data reports or event/trap reports with the SunNet Manager Results Browser and Results Grapher. (See Chapter 20, “Browser” and Chapter 21, “Results Grapher,” for more information.)

This chapter discusses the following topics:

- Props button for setting global preferences
- Props button categories:
 - Windows
 - Requests
 - Automatic Management
 - Events and Traps
 - Errors
 - Locations
 - Customizable colors
 - Miscellaneous
 - Other Configurations

Use the Console Properties button to configure global preferences for the operation of the Console through a window interface. In addition, you can define certain operations in a Console resource file—see Section 17.10, “Other Configuration,” on page 17-29 for more information.

17.1 Global Properties

You can set global properties from the Console Props button in the Console Properties window. You display the Console properties window by ensuring no glyphs are selected in the Console window (click SELECT anywhere in the Console window that is not a glyph), then clicking SELECT on the Props button as illustrated in Figure 17-1.

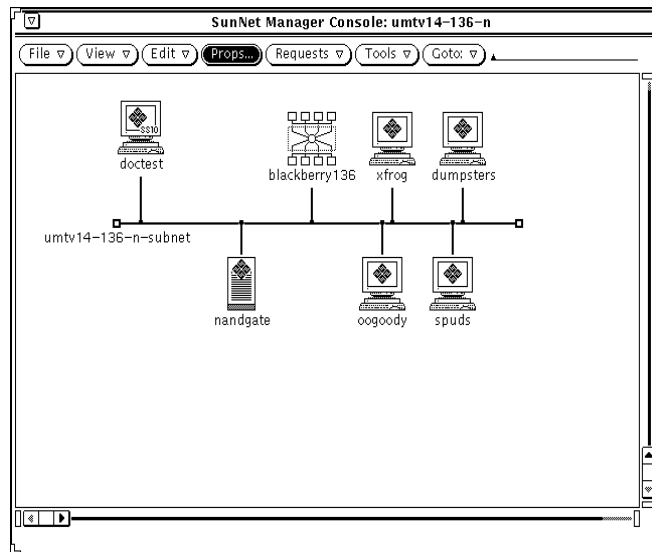


Figure 17-1 Selecting Props Button

Clicking on the Props button brings up the window shown in Figure 17-2.

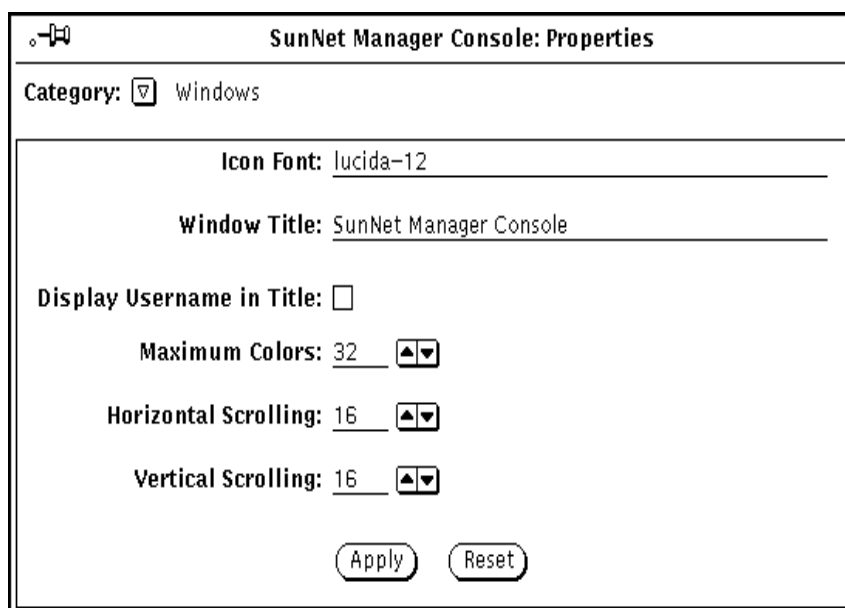


Figure 17-2 Console Properties Window

The Console Properties window has a Category abbreviated menu button that provides the window categories shown in Figure 17-3.

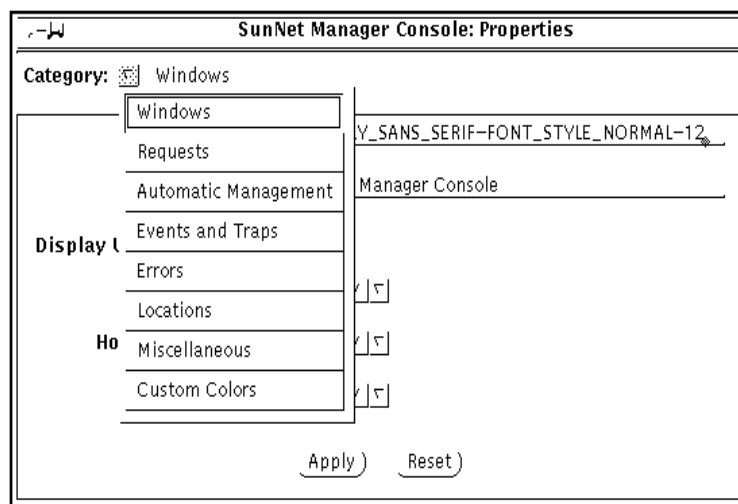


Figure 17-3 Console Properties Window Categories

These categories are described in the following subsections. When you choose a category from the menu, the name to the right of the abbreviated menu button changes to reflect the new category, and the property window displays a pane with the settings for the selected category.

After you have entered in all the changes in a window, click **SELECT** on the **Apply** button to apply the new settings and dismiss the Console Properties window. Click **SELECT** on the **Reset** button to return to the previous settings. Click **SELECT** on the pushpin in the upper right corner to unpin, and thereby dismiss, the Console Properties window.

Most changes to window settings take effect during your current Console session, although you may not see the change immediately. For instance, if you change the window title, you will not see the change until you change views or the currently-displayed Console window is changed. A few changes to window settings do not take effect until you restart the Console—these settings are noted in the descriptions in Section 17.2, “Windows.”

17.2 Windows

The settings in the Windows category define the properties of the main windows of the Console and Console tools. The items in the Console Properties window for the Window category are as follows:

Icon Font:

Specifies the font used by the Console to display the labels under the glyphs. The default is “lucida-10.” You can enter a font that is available on your server—use the `xlsfonts` command to see available fonts.

Note – You must restart the Console in order for changes to the Icon Font setting to take effect.

Window Title:

Specifies the text that appears in the title bar of *all* of the Console’s windows. The default is “SunNet Manager Console.” The text can be up to 80 characters (no quotation marks are required). If the text is too long for the size of a window, only a view name is displayed.

Display Username in Title:

Specifies whether or not the value of the environment variable `SNM_USER` (or `USER` if `SNM_USER` is not set) is displayed in the title bar in parentheses just after the value of Window Title. A check mark in the box indicates that the user name is to be displayed in the title bar—click **SELECT** on the box to toggle the check mark on or off. By default, this is off (no user name is displayed).

Maximum Colors:

Specifies the number of user colormap entries the Console uses. This value is the maximum number of unique colors which can be supported in a given view. The default is 32. To change the value, either type in a new number or click **SELECT** on the down arrow. The range of possible values is 16 to 255.

Note – You must restart the Console in order for changes to the Maximum Colors setting to take effect.

Horizontal Scrolling:

Specifies the number of pixels to move whenever you click SELECT on the left or right arrows of the horizontal scroll bar of a Console window. The default is 16. To change the value, either type in a new number or click SELECT on the up or down arrows. The range of possible values is 1 to 99.

Vertical Scrolling:

Specifies the number of pixels to move whenever you click SELECT on up or down arrows of the vertical scroll bar of a Console window. The default is 16. To change the value, either type in a new number or click SELECT on the up or down arrows. The range of possible values is 1 to 99.

17.3 Requests

The settings in the Requests category define certain Data and Event Request properties. The Requests category is shown in Figure 17-4, “Console Properties Requests Category.”

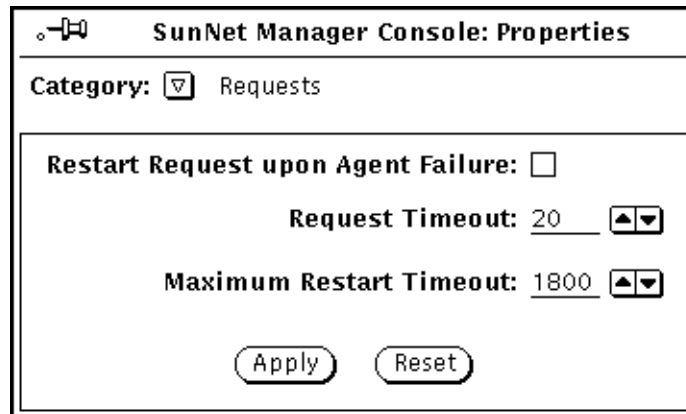


Figure 17-4 Console Properties Requests Category

Restart Request upon Agent Failure:

Specifies the default value of the Restart field in the Request Properties window. A check mark in the box specifies that whenever a new Request Properties window is displayed, the Restart field in the Request Properties window is on (a check mark appears next to the Restart field). The default value can be overwritten in the Request Properties window for individual

requests. Click **SELECT** on the box to toggle the check mark on or off. By default, this field is off (the Restart field is off in the Request Properties window).

Request Timeout:

Specifies the time (in seconds) that the Console waits for a response from an agent when starting or stopping requests. The default value is 20. The range of possible values is 10 to 300 (5 minutes).

Maximum Restart Timeout:

Specifies the maximum interval between attempts to start a failed request. If a request fails to start, the Console retries the request after 60 seconds. If the request fails again, it is retried again after 120 seconds. The interval between failed requests and the re-sending of the request is doubled for each restart attempt. If the maximum restart timeout value is reached before the agent responds to the request, the maximum restart timeout value becomes the interval between resent requests. The default value is 1800 seconds (30 minutes). The range of possible values is 120 to 3600 (1 hour).

17.4 Automatic Management

You can enable Automatic Management to automatically monitor the health of your network. You do this by sending an event request to each managed device to determine the state of the device. This feature also allows predefined Event Requests to start automatically for elements that are added to the management database. The Automatic Management category in the Console Properties windows is shown in Figure 17-5.

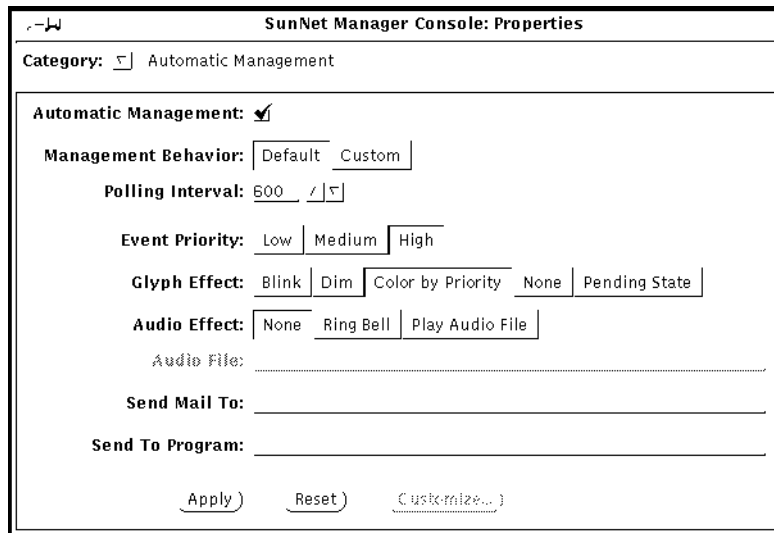


Figure 17-5 Console Properties Automatic Management Category

By default, the automatic node management feature is disabled. If you enable this feature, you can also define the reporting characteristics and signal options of automatic requests. Before you enable automatic node management, read the following section, which describes how it works.

If you enable automatic node management, you can specify that predefined Event Requests *not* be sent for elements that are created in certain views or buses. This is done with the Auto Manage Off option in the bus or view's Glyph menu. Refer to Section 14.10.12, "Auto Manage Off," on page 14-32 for more information about this option.

If you enable automatic node management and then load an ASCII database file, the Console will attempt to start a predefined Event Request for every element defined in the database that supports one of the proxy agents listed above. You can disable automatic node management for certain views. This is done with the Auto Manage Off option of the element's Glyph menu (refer to Section 14.10.12, "Auto Manage Off," on page 14-32 for more information). You can kill launched Event Requests at the same time by using the Requests Summary window. Once you have killed these requests, re-save the database to an ASCII file. The next time the file is loaded, no predefined Event Requests will be started.

You can use automatic management in one of two modes: default or custom. Choose the mode (behavior) by selecting the appropriate value in the management behavior field as shown in Figure 17-5.

If you choose Default mode, SunNet Manager will send predefined requests as explained in Section 17.4.1, “The Default Option.”

17.4.1 The Default Option

If automatic node management is enabled with the default option, a predefined Event Request is automatically started for new elements that support at least *one* of the following proxy agents:

- SNMP
- hostperf
- ICMP ping

The event specification for each default proxy agent is described below.

If the element supports more than one of the proxy agents in default mode, the predefined Event Request that is actually sent is determined in the following order:

1. If you have specified that the new element can be managed with a Sun-supplied SNMP schema, the following is defined as the event for the element:

```
ifOperStatus Not Equal To 1
```

which specifies the condition where a device is not up.

If the element can be managed with multiple Sun-supplied SNMP schemas, the Console uses the following order of schema files:

- a. `snmp.schema` (MIB I)
- b. `snmp-mibII.schema` (MIB II)
- c. `sun-snmp.schema` (MIB II with Sun enterprise-specific extensions for Sun workstation support)

For example, if an element supports MIB I as well as the Sun workstation MIB, the Console uses the schema file associated with MIB I (`snmp.schema`) for the auto request. Note that if the element supports SNMP but does not support the `ifOperStatus` attribute, an error is

returned to the Console. If this happens, the request is stopped. If the element supports the `hostperf` proxy agent, the predefined Event Request that is associated with the `hostperf` proxy agent is sent. Otherwise, if the element supports the `ping` proxy agent, the predefined Event Request that is associated with the `ping` proxy agent is sent.

2. If you have specified that the new element can be managed with the `hostperf` agent, the following is defined as the event for the element:

uptime increased by less than *<number>*

where *<number>* is the value of the Polling Interval setting in the Console Properties Automatic Management category minus 2 minutes. By default, the value of *<number>* is 8 minutes.

3. If you have specified that the new element can be managed with the `ping` agent, the following is automatically defined as the event for the element:

reachable equal to 'false'

Once started, an automatic Event Request can be modified like any other request. The window for the Automatic Management category is shown in Figure 17-5 on page 17-8, with the automatic node management feature enabled (checked off). If you disable the automatic node management feature, the rest of the items in the window are grayed-out.

The items in the Automatic Management category window are described below:

Automatic Management:

Specifies whether the automatic node management feature is enabled. A check mark in the box indicates that the feature is enabled—click SELECT on the box to toggle the check mark on or off. When this field is off (automatic node management is disabled), the remaining items in the window are dimmed. If automatic node management is not enabled (check box is empty and you click SELECT on Apply), the Auto Manage option in the element Glyph menus is not available. Note that if you enable automatic node management then later disable it, all active automatic requests are killed.

Polling Interval:

Specifies the interval (in seconds) for the agent to send reports. By default, the reporting frequency for automatic requests is set to '600' (10 minutes). To change the value, either type in a number on the line or click SELECT on the up or down arrows. The range of possible values is 1 to 9999 (seconds).

Caution – If you decrease this number substantially, you can cause network traffic problems.

Event Priority:

Specifies the priority of an event from an automatic node management request. The default is High. To change the setting, click SELECT on the appropriate rectangle.

Glyph Effect:

Specifies a visible indicator when an event is reported from an automatic node management request. Only one glyph effect option can be chosen. The default is Color by Priority. To change the setting, click SELECT on the appropriate rectangle.

Audio Effect:

Specifies an audible indicator when an event is reported from an automatic node management request. Only one audio effect option can be chosen. The default is None (no audible indicator). To change the setting, click SELECT on the appropriate rectangle.

Audio File:

Defines the audio file to be played when an event is reported from an automatic node management request. (The option Play Audio File must be specified in the Audio Effect field for this setting to be enabled.) Type in the path name of the audio file. The audio file will be played at the currently-set audio level.

Note – The Console must be running (not merely displaying) on a machine with an audio port. If you are running the Console on a server but displaying the Console windows on a local workstation, the Console will attempt to play the audio file on the server.

Send Mail To:

Defines one or more mail recipients to whom an Event Report is sent when an event is reported from an automatic node management request. Type in a list of mail recipients. If there is more than one recipient, use a space or comma between each entry.

Send To Program:

Defines a program or shell script to be run when an event is reported from an automatic node management request. Include the directory path, if necessary. A new copy of the program or shell script is forked for each event report. The event report is passed to the standard input of the program or shell script. For example, if the value were `cat > /tmp/trap.rpt`, it would cause the `cat` program to receive the event report and write it to the file `/tmp/trap.rpt`.

17.4.1.1 The Custom Option

Starting with version 2.3, SunNet Manager provides a powerful addition to Automatic Management called customized auto management. This feature allows you to selectively perform auto management on a particular class of devices.

If the default action does not meet your requirement, you can create a predefined request by selecting the Custom option in the management behavior field as shown in Figure 17-5. Your predefined requests will be attached with the type of node against which you want to run the request. Click SELECT on Customize to see the Customize Automatic Management window as shown in Figure 17-6.

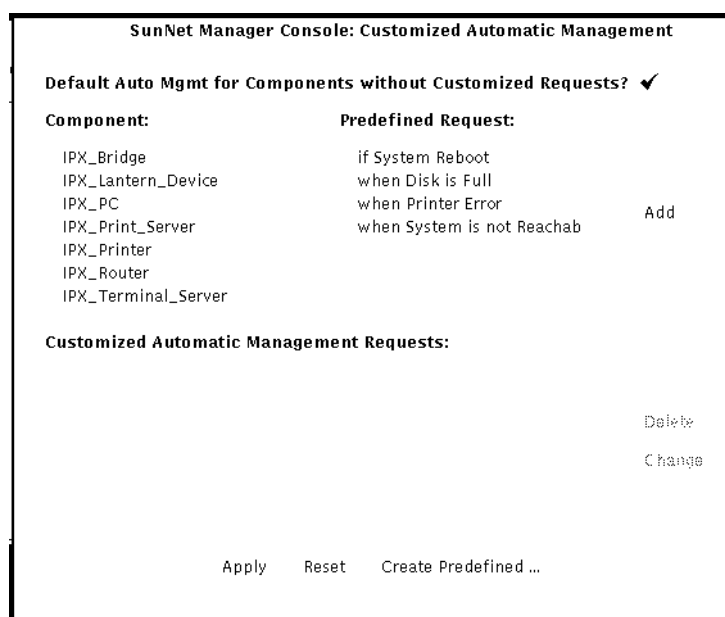


Figure 17-6 Automatic Management Customize Window

The Component list on the left side of the screen will list the components in your runtime database that were identified during initialization from the `elements.schema` file. The Predefined Request list is read from the `SNMpredefined` file in your home directory.

A predefined request can be attached to a component type using the following steps. Refer to Figure 17-7 for examples of the fields explained in the steps below.

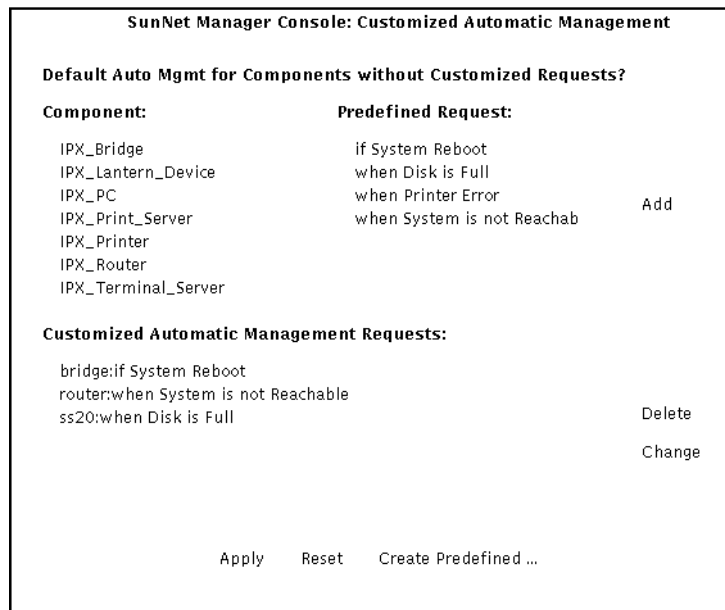


Figure 17-7 Customizing Auto Management

1. Use the mouse to select a component from the component list.
2. Use the mouse to select a request from the predefined request list.
To create a predefined request, click SELECT on Create Predefined and see Chapter 5, “Specifying Event Requests” for information on creating predefined event requests.
3. Click SELECT on the Add button.

The component/request pair appear in the Customized Automatic Management Request list.

You can detach a predefined request from the component type using the following steps:

1. Use the mouse the select the desired entry from the Customized Automatic Management Request list.
2. Click SELECT on Delete.

You can also detach a predefined request from a component and attach a new predefined request using the following steps:

1. Use the mouse to select the desired entry from the Customized Automatic Management list.

The corresponding component/predefined requests are automatically selected.

2. Use the mouse to select a new predefined request from the Predefined Request list.

3. Click SELECT on the Change button.

4. Click SELECT on Apply to submit changes.

Note that if an element or component supports more than one agent, only *one* predefined Event Request is sent.

The changes are saved to `.SNMautomanagement` in your home directory and the request(s) is launched.

You can combine customized automatic managed with default automatic management. This means you can choose to run predefined requests for a specific component type (such as a router), and at the same time run default automanagement for the rest of the nodes. To combine the features:

1. Select a component from the component list.

2. Select a predefined request from the Predefined Request list.

3. Click SELECT on the Add button.

4. Click SELECT on the checkbox at the top of the customized window as shown in Figure 17-6.

5. Click SELECT on Apply.

SunNet Manager will run your predefined request for the component you selected and will run default automatic management for the rest of the components.

To run only customized requests, do not click the checkbox at the top of the customized window.

17.5 Events and Traps

The settings in the Events and Traps category allow you to define Console operations for events and traps. You can also specify signal options for trap reports.

Note – Trap report signal options are effective for all trap reports the Console receives.

The window for the Events and Traps category is shown in Figure 17-8.

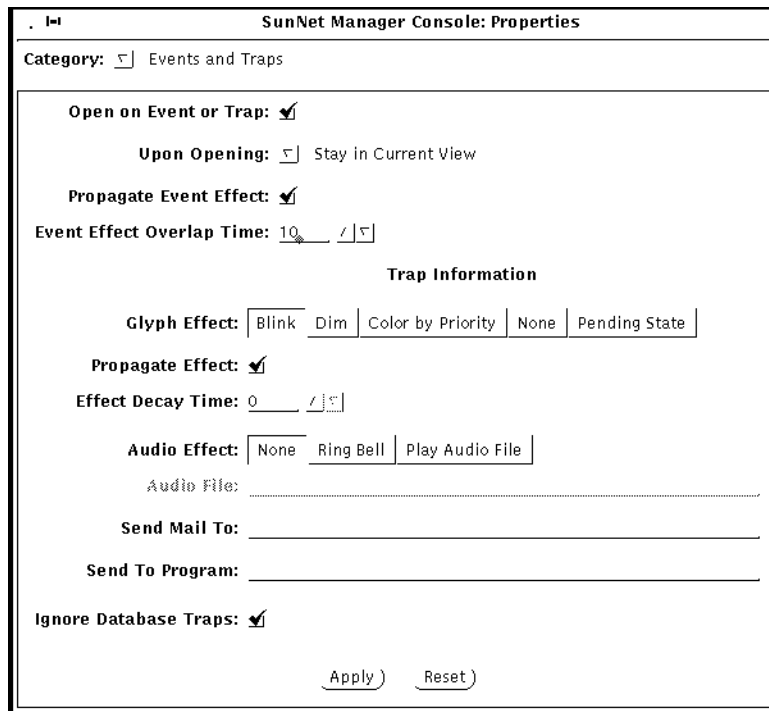


Figure 17-8 Console Properties Events and Traps Category

Open on Event or Trap:

Specifies whether the Console window should be opened if it is closed to an icon when an event or trap report is received. A check mark in the box indicates that Console should be opened—click SELECT on the box to toggle the check mark on or off. By default, this is off (Console window is not opened).

Upon Opening:

The Open on Event or Trap setting described above *must* be checked for this setting to be enabled. Upon Opening specifies the view to be displayed when the Console window is opened and an event or trap report has been received. Press MENU in the Upon Opening abbreviated menu button to receive the menu in Figure 17-9.

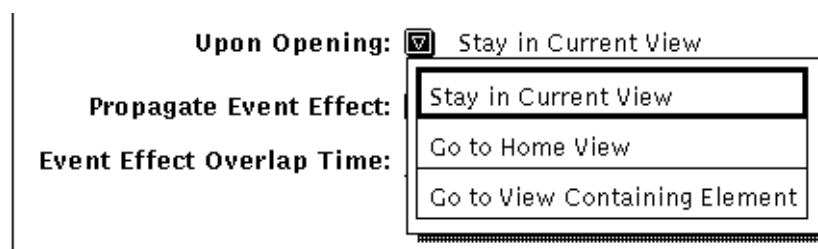


Figure 17-9 Events and Traps—Upon Opening Menu

The default is Stay in Current View (the Console displays the same view it displayed before the Console window was closed to an icon). To change the setting, press MENU on the abbreviated menu button and release on the desired option. Go to Home View causes the Console to always display the Home view while Go to View Containing Element causes the Console to display the view that contains the element for which an event or trap report is received.

Propagate Event Effect:

Specifies whether the glyph effect options of events are propagated to all parent views of the affected element. A check mark in the box indicates that propagation is enabled—click SELECT on the box to toggle the check mark on or off. By default, this is on (events' effects are propagated through the view hierarchy).

Note – You can disable glyph state propagation for a view by setting the Glyph State field in the view’s Property window to “Not Inherited.”

If a glyph is contained in more than one view, its state is propagated through its multiple view hierarchies. When the Console is closed to an icon, glyph state changes reflected in the Home view cause the SNM Console icon to display question marks.

If there are different glyph states propagating to the same view glyph, the view glyph assumes the state of the highest-priority member glyph state. The following list defines glyph-state priority from highest to lowest:

- a. Color set by the priority of the event. By default, high, medium, and low priority events are red, orange, and yellow, respectively.
- b. Dimmed
- c. Blinking

You can explicitly change the state of a glyph by using the Glyph State option in the Glyph menu for the element. There are two important points to remember when using this option to change glyph state:

- When you change the state of a glyph, the new glyph state is propagated as though an event had occurred to change the glyph state. For example, if you change the glyph state of an element from normal to blinking, the glyph state (blinking) propagates as if an event had occurred.
- If you have multiple glyph states propagating into a single view, and you change the glyph state for the view to normal, the member glyph states are also reset to normal. For example, if you change the glyph state of a view from blinking to normal, the effects of the event that caused a member element to blink are cleared. This allows you to clear multiple events by resetting the glyph state of a view.

Starting with version 2.3, you can specify pending mode for a glyph state, (press MENU over the target glyph; release MENU over Pending On/Off). The glyph object is dimmed and outstanding traps/events are cleared. New traps/events do not change the glyph state or propagate the effect of the trap/event to the parent object. If a parent object is in pending state, a state change for any child object has no affect on the parent. See Chapter 5, “Specifying Event Requests,” for more information about glyph pending state.

Event Effect Overlap Time:

Specifies the amount of time (in seconds) beyond the reporting interval that an event's glyph effect options remain in effect. If another event does not occur within this overlap time, the Console cancels the current signal option. The default is 10 seconds. To change the value, either type in a number on the line or click SELECT on the up or down arrows. The range of possible values is 0 (no overlap) to 86400 (24 hours).

Note – If the reporting interval is not specified in the event report, the Console assumes the reporting interval to be 60 seconds.

The following fields pertain to traps only.

Glyph Effect:

Specifies a visible indicator when a trap is reported. The default is Color by Priority. Starting with version 2.3, you can customize the color of the glyph to indicate Low, Medium or High priority by clicking on Props►Custom Colors on the Console menu as shown in Figure 17-10.

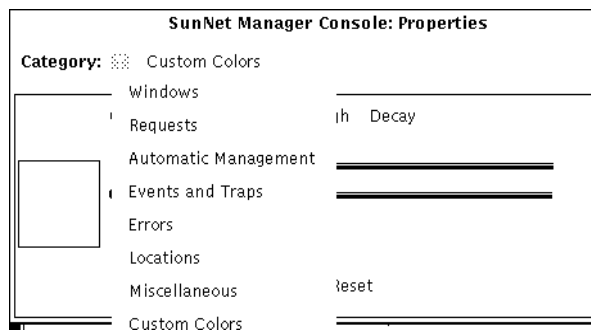


Figure 17-10 Custom Colors Category of Console Props Menu

When you receive the Custom Colors window in Figure 17-11, use the mouse to slide the indicators to the color you wish for each priority level.

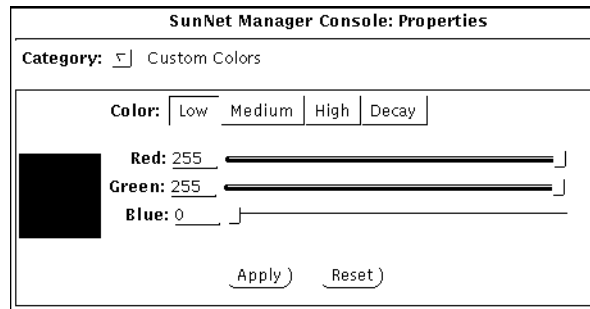


Figure 17-11 Custom Color Window

Note that the Console can receive high, medium, and low priority traps. (Low priority traps are generated when the SNMP database API functions are used to add, change, or delete elements in the database.) If you keep the default Color by Priority option, high priority traps cause the element's glyph to turn red and low priority traps cause the glyph to turn yellow. If both high and low priority traps are received for an element, high priority is used. In association with the Color by Priority option is the decay feature. The decay feature pertains to traps that have been reported and for which the Effect Decay Time period has expired. In response to these conditions, the glyph for the affected element turns blue (the default), or to a color you have customized. Once a glyph shows the decay color, the trap report must be acknowledged in order for the glyph to return to its original color. To get a glyph to return to its original color, you can either:

- Select the Console's View►Event Summary option
- Highlight the name of the glyph
- Press the Drop from List button.
- Or
- From the Glyph Menu for the specific element, pull right over the Glyph State►Normal option.

Propagate Effect:

Specifies whether the glyph effect options of traps are propagated to all parent views of the affected element. A check mark in the box indicates that propagation is enabled—click SELECT on the box to toggle the check mark on or off. By default, this is on (trap’s effects are propagated through the view hierarchy).

Glyph state changes that result from traps propagate in the Console’s view hierarchy in the same manner as with glyph state changes that result from events. See the description of the Propagate Event Effect setting for more information about propagation of glyph state changes.

Effect Decay Time:

Specifies the amount of time (in seconds) during which a trap’s glyph effect option remains in effect. After the specified time, the effect caused by the trap is automatically cleared and the glyph’s state is returned to “Normal” or decayed to blue. The latter only occurs when the Color by Priority option is in effect as described above under Glyph Effect. The default is 0 (glyph effects caused by a trap are not automatically cleared). To change the value, either type in a number on the line or click SELECT on the up or down arrows. The range of possible values is 0 to 86400 (24 hours).

Audio Effect:

Specifies an audible indicator when a trap is reported. Only one audio effect option can be chosen. The default is None (no audible indicator). To change the setting, click SELECT on the appropriate rectangle.

Audio File:

Defines the audio file to be played when a trap is reported. (The option Play Audio File must be specified in the Audio Effect field for this setting to be enabled.) Type in the path name of the audio file. The audio file will be played at the currently-set audio level.

Note – The Console must be running (not merely displaying) on a machine with an audio port. If you are running the Console on a server but displaying the Console windows on a local workstation, the Console will attempt to play the audio file on the server.

Send Mail To:

Defines one or more mail recipients to whom a trap report is sent when a trap is reported. Type in a list of mail recipients. If there is more than one recipient, use a space or comma between each entry.

Send To Program:

Defines a program or shell script to be run when a trap is reported. Include the directory path, if necessary. A new copy of the program or shell script is forked for each trap report. The trap report is passed to the standard input of the program or shell script. For example, if the value were `cat > /tmp/trap.rpt`, it would cause the `cat` program to receive the trap report and write it to the file `/tmp/trap.rpt`.

Ignore Database Traps:

Defines whether the traps generated when changes are made in the database are displayed by the Console. Low priority traps are generated when elements are added, changed, or deleted in the database, or if a new database file is loaded. A check mark in the box indicates that these traps are ignored by the Console — click SELECT on the box to toggle the check mark on or off.

Note – With the Ignore Database Traps option turned off, when you create an element, the trap report for that element says that the element was “created.” An analogous operation using the database API returns the word “added” rather than “created.”

17.6 Errors

The settings in the Errors category allow you to specify signal options for errors that the Console receives from agents. Note that the only type of errors that cause glyph state changes are errors received from agents; glyph state changes do not occur as a result of local Console errors. The window for the Errors category is shown in Figure 17-12, “Console Properties Errors Category.”

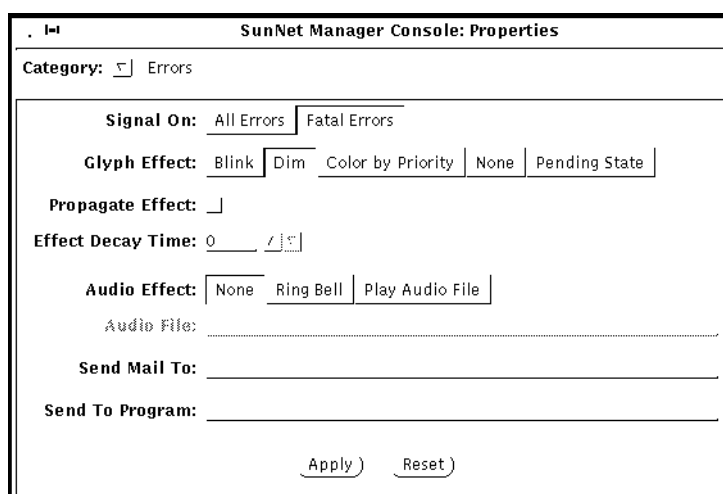


Figure 17-12 Console Properties Errors Category

Signal On:

Specifies whether signal options are in effect for all errors received from agents or only fatal errors. The default is Fatal Errors. To change the setting, click SELECT on All Errors.

Glyph Effect:

Specifies a visible indicator when an agent error is reported. Only one glyph effect option can be chosen. The default is Dim. To change the setting, click SELECT on the appropriate rectangle. Note that agent errors can be either high or low priority errors. If you choose the Color by Priority option, high priority errors (generic errors and agent-specific fatal errors) cause the element's glyph to turn red and low priority errors (agent-specific warnings) cause the glyph to turn to the default color of yellow. If both high and low priority errors are received for an element, high priority is used.

Propagate Effect:

Specifies whether the glyph effect options of errors are propagated to all parent views of the affected element. A check mark in the box indicates that propagation is enabled—click SELECT on the box to toggle the check mark on or off. By default, this is off (error’s effects are not propagated).

Glyph state changes that result from errors propagate in the Console’s view hierarchy in the same manner as with glyph state changes that result from events. See the description of the Propagate Event Effect setting in the Events and Traps category for more information about propagation of glyph state changes.

Effect Decay Time:

Specifies the amount of time (in seconds) during which an error’s glyph effect option remains in effect. After the specified time, the effect caused by the error is automatically cleared and the glyph’s state is returned to “Normal.” The default is 0 (glyph effects caused by an error are not automatically cleared). To change the value, either type in a number on the line or click SELECT on the up or down arrows. The range of possible values is 0 to 86400 (24 hours).

Audio Effect:

Specifies an audible indicator when an error is reported. Only one audio effect option can be chosen. The default is None (no audible indicator). To change the setting, click SELECT on the appropriate rectangle.

Audio File:

Defines the audio file to be played when an error is reported. (The option Play Audio File must be specified in the Audio Effect field for this setting to be enabled.) Type in the path name of the audio file. The audio file will be played at the currently-set audio level.

Note – The Console must be running (not merely displaying) on a machine with an audio port. If you are running the Console on a server but displaying the Console windows on a local workstation, the Console will attempt to play the audio file on the server.

Send Mail To:

Defines one or more mail recipients to whom an error report is sent when a error is reported. Type in a list of mail recipients. If there is more than one recipient, use a space or comma between each entry.

Send To Program:

Defines a program or shell script to be run when an error is reported. Include the directory path, if necessary. A new copy of the program or shell script is forked for each error report. The error report is passed to the standard input of the program or shell script. For example, if the value were `cat > /tmp/trap.rpt`, it would cause the `cat` program to receive the error report and write it to the file `/tmp/trap.rpt`.

17.7 Locations

The settings in the Locations category allow you to define the directories where the schema and icon files used by the Console are found. The Locations category is shown in Figure 17-13, “Console Properties Locations Category.”

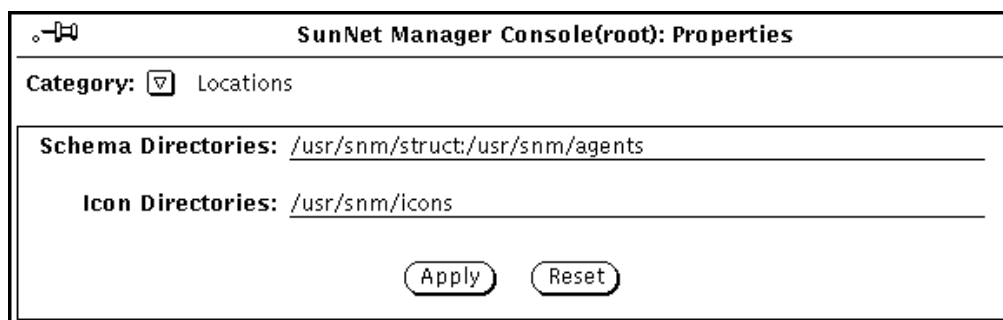


Figure 17-13 Console Properties Locations Category

Note – You must restart the Console in order for changes to the Locations settings to take effect.

Schema Directories:

Specifies one or more directories that contain agent and element schema files. The Console searches these directories from left to right, loading any files ending in `.schema`. Multiple directories are separated by colons (`:`). The directories must be absolute path names (start with `/`) and cannot contain environment variables.

Icon Directories:

Specifies one or more directories that contain icon and icon mask files. The Console searches these directories from left to right, loading any files ending in `.icon` and `.iconmask`. Multiple directories are separated by colons (:). The directories must be absolute path names (start with /) and cannot contain environment variables.

17.8 Miscellaneous

The settings in the Miscellaneous category allow you to define Console parameters not covered in the other categories. The window for the Miscellaneous category is shown in Figure 17-14, “Console Properties Miscellaneous Category.”

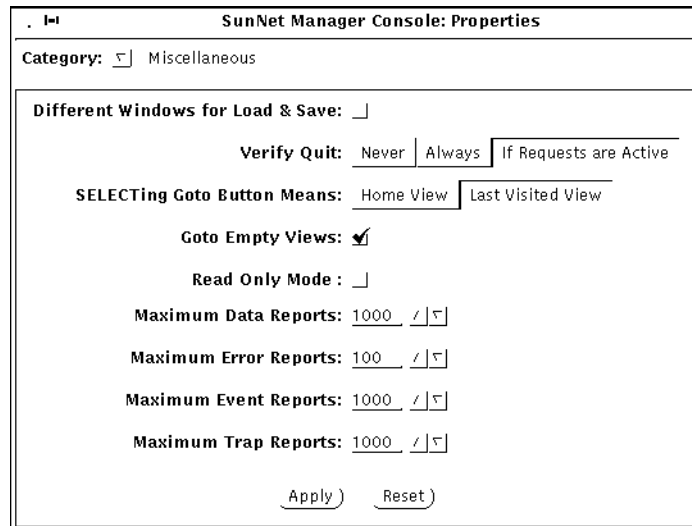


Figure 17-14 Console Properties Miscellaneous Category

Different Windows for Load & Save:

Specifies whether different windows are displayed for Load and Save operations. A check mark in the box indicates that different windows are displayed—click SELECT on the box to toggle the check mark on or off. By default, this is off (the same window is used for Load or Save operations).

Verify Quit:

Specifies the verification prompt, if any, displayed by the Console whenever you choose to quit a Console session. The default is If Requests are Active (a verification prompt is displayed only if there are active requests; the verification prompt asks if you want the active requests automatically killed before the Console is exited). To change the setting, click SELECT on the appropriate rectangle: Never causes no verification prompt to be displayed, Always causes a verification prompt to be displayed even if there are no active requests.

SELECTing Goto Button Means:

Specifies the view that is displayed whenever you click SELECT on the Goto menu button. The default is Last Visited View. To change the setting, click SELECT on Home View (the Home view is displayed whenever you click SELECT on the Goto menu button).

Goto Empty Views:

Specifies whether or not the Console displays an empty view. A check mark in the box indicates that the Console can display empty views. Click SELECT on the box to toggle the check mark on or off. By default, this is on (Console can display empty views). If the box is not checked, the Console does not display empty views and an error message appears in the Console's footer. However, an empty view can still be selected if the user clicks SELECT twice on the empty view.

Read-Only Mode

In read-only mode, you can run your applications and manage requests. However, the following menu items are dimmed: Edit, Create Predefined Requests, and Change Type. See Chapter 14, "Console" for more information regarding Read-Only mode.

Maximum Data Reports:

Defines the maximum number of data reports that can be displayed in the Data Reports window. The default is 1000. The range of possible values is 10 to 99999.

Maximum Error Reports:

Defines the maximum number of error reports that can be displayed in the Error Reports window. The default is 100. The range of possible values is 10 to 99999.

Maximum Event Reports:

Defines the maximum number of event reports that can be displayed in the Event/Trap Reports window. The default is 1000. The range of possible values is 10 to 99999.

Maximum Trap Reports:

Defines the maximum number of trap reports that can be displayed in the Event/Trap Reports window. The default is 1000. The range of possible values is 10 to 99999.

17.9 Custom Colors

Starting with version 2.3, you can customize trap priority colors by selecting Props►Category►Custom Colors. You receive the screen in Figure 17-15.

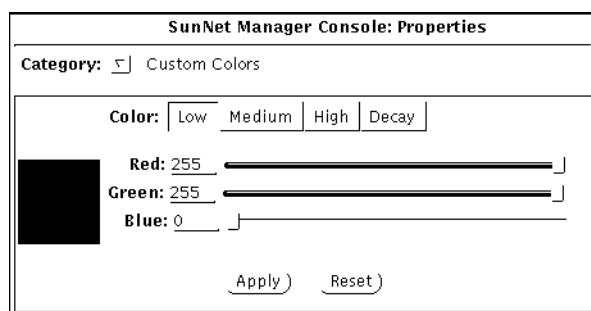


Figure 17-15 Custom Colors Window

See Section 17.5, “Events and Traps,” on page 17-16 for a description of the Custom Colors function.

When you click SELECT on Apply after selecting the color(s) you wish, the changes are saved to the `.SNMdefaults` file. If the colors are different from the previous values, the glyph state colors for the target objects are recalculated and displayed.

17.10 Other Configuration

When you apply a setting in the SunNet Manager Console Properties window (under Props, when no glyphs are selected), the information is stored in the `$HOME/.SNMdefaults` file. The `$HOME/.SNMdefaults` file is intended to contain per-user configuration information.

Per-product defaults can be placed in `$$SNMHOME/defaults`. This allows you to set up site-wide Console resources and definitions so that everyone who uses the product uses those defaults. If the user has their own `$HOME/.SNMdefaults` file, the information in the `$HOME/.SNMdefaults` overrides the information in `$$SNMHOME`. Users can also set the environment

variable `SNMDEFAULTS` to have the Console read another file instead of `$HOME/.SNMdefaults`. If `SNMDEFAULTS` is not set, the Console reads `$HOME/.SNMdefaults`.

Note – The Console also supports X11 resources configurations, however settings in `$HOME/.SNMdefaults` take priority.

There are a few optional definitions you can specify to further customize the operation of the Console. You can add these definitions into one of the files described above.

17.10.1 Forking Programs For Element Types

You can specify a program that is forked for an element type when an instance of the element type is created or modified. To do this, use the following definitions:

```
snm.console.<element-category>.<element-type>.createProg: <program1>  
snm.consolecomponent.<element-category>.<element-type>.modifyProg: <program2>
```

where `<program1>` and `<program2>` are forked whenever the `<element-type>` is created or modified, respectively. Note that you can specify `*` for `<element-category>` or `<element-type>`. If you want programs to be forked for any creation or modification of an element, use the following definitions:

```
snm.console.*.*.createProg: <program1>  
snm.console.*.*.modifyProg: <program2>
```

If you want a program to be forked whenever any view is created, use the following definition:

```
snm.console.view*.createProg: <program>
```

If you want a program to be forked whenever an ethernet element type (`bus.ethernet`, `view.ethernet`, etc.) is created, use the following definition:

```
snm.console*ethernet.createProg: <program>
```

The process that is forked when an element is created has the following parameters:

```
<program1> -c -v <viewname> -t <object_type>
```

where `<viewname>` is the current view, and `<object_type>` is the type of the object created.

The process that is forked when an element is modified has the following parameters:

```
<program2> -m -v <viewname> -n <object_name>
```

where `<viewname>` is the current view, and `<object_name>` is the name of the object being modified.

17.10.2 Redirecting SNMP Requests

You can have SNMP requests from the Console redirected to another vendor's SNMP proxy. To do this, use the following definition:

```
snm.console.snmpRedirect: true  
snm.console.snmpRedirectItem: <label>  
snm.console.snmpRedirectAgent: <agent>
```

where `<label>` appears in the bottom left panel of the Data and Event Request Properties window and `<agent>` is the name of the SNMP proxy agent to which the requests will be redirected. In the Request Properties window, a box appears next to `<label>`—click SELECT in the box to toggle a check mark on or off. A check mark causes the request to be redirected.

Note – The Console does not kill redirected requests when it starts or exits. You must explicitly kill these requests with the `snm_kill(1)` command. This feature is intended for vendor-supplied SNMP proxies that do not return reports to the Console.

17.10.3 Activating Console Database Manager Traps for File Loads

The Console can be configured to issue a trap to inform other applications when a new runtime database is loaded. Such a trap contains the names of the elements that have been loaded.

To activate this feature, the `.SNMdefaults` file needs to be modified to set the value of `snm.console.DBMgrTrapAlways` to true. By default, this property is set to false.

17.11 Custom Colors

Starting with version 2.3, you can customize colors of event and trap priority levels. The default colors continue to be Low=yellow, Medium=orange, High=Red, and decay=blue. By choosing the Custom Colors category, you receive the window in Figure 17-16.

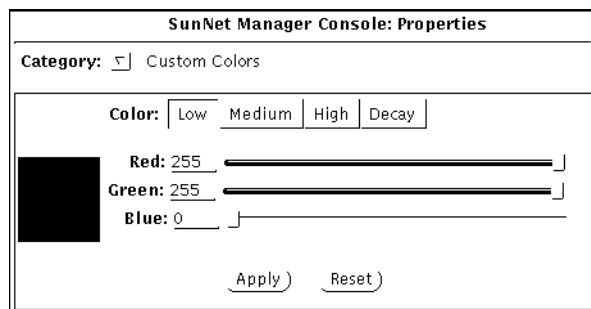


Figure 17-16 Custom Colors Configuration Window

You can determine the current RGB value of a color for particular objects by choosing the checkbox corresponding to the priority state or the decay state of the element. Customize the color by sliding the color indicator(s) to the desired position. If the colors are different from the previous values, the glyph state colors for the objects are recalculated and displayed.

Click SELECT on Apply to save the new color to the `.SNMdefaults` configuration file.

This chapter discusses the following topics:

- Element type definitions
- Element instance definitions
- Connection definitions
- Background definitions
- Tools menu definitions
- Definition of requests
- Duplicate databases

18.0.1 Purpose of the Management Database

The management database (also called the runtime database) represents the data model of a site's environment. It contains information on the agents that are supported and the objects that are managed. The information in the management database determines how the SunNet Manager Console displays the layout of the site's network, machine icons (glyphs in OPEN LOOK parlance), characteristics, and associated menus.

Certain kinds of changes to your management database can *only* be made by editing a saved MDB file with a text editor. These kinds of changes include:

- Adding or modifying element types.

- Specifying a new glyph for an element type.
- Adding or modifying tools in an element's Glyph menu.

“Part 1: Network Management Tasks” describes how to make these changes. This chapter describes the format of the definitions contained in the database files. Specifically, these definitions include:

- Element type, including definition of the glyph and user commands associated with an element type.
- Element instances.
- Connections between elements.
- Background images for view instances.
- Tools available from the Console's Tools menu.
- Data requests.
- Event requests.

Input to the runtime database comes from two sets of ASCII files, *structure* files and *instance* files.

Structure files provide a set of definitions to the Console. These include:

- The `snm.glue` file that contains a starting set of definitions for the Console. The Console depends on these definitions, so do **not** modify them.
- The agent schemas that define the attributes each agent manages. (See your *Site/SunNet/Domain Manager Application and Agent Development Guide* for a discussion of agent schema files.)
- The element schema file(s) that define the element types in a site's configuration. Element schema also specify the glyphs and commands for each element type. SunNet Manager provides definitions of numerous element types in the file, `elements.schema`. See Section 18.1, “Element Type Definition,” for more information.

Note – Do *not* change the `elements.schema` or `netware_elements.schema` file. If you need to add element types, create a file with the `.schema` extension in the same directory as the `elements.schema/netware_elements.schema`.

Instance files provide definitions of element and request instances.

The runtime database can also be modified by the Console through its graphical editing capability and through the Discover Tool. As you move glyphs and start requests, you are modifying the runtime management database.

18.1 Element Type Definition

Although many element types are defined in the file, `elements.schema`, additional types may be added or current types modified to customize the Console to meet a particular site's needs. This section discusses element type definition.

Note – Element types are constructed from data types specified in the enumerated type `snmdb_type`, which is defined in the header file `netmgt_db.h`. These data types are listed and described in your *Site/SunNet/Domain Manager Application and Agent Development Guide*.

There are four categories of elements. New element categories cannot be added by a user. The element categories are:

- **Components.** Components represent elements such as workstations, printers, and routers. Components appear as a glyph (see Section 18.1.1, “Glyph Definition (for Component or View Element Type),” for a discussion of how to specify the glyph for a component).
- **Views.** A view is a collection of elements. A view may contain other views. Like a component, a view also appears as a glyph.
- **Buses.** Buses represent elements such as an Ethernet LAN segment. A bus appears as a line with movable endpoints.
- **Connections.** A connection is an element that connects two other elements. A connection appears as a line connecting the two elements. Examples are a leased internet work link and an RS-232-C line.

Element types are defined using a *record*. The record specifies the fields stored in the database for each instance of that element type. The format is:

```
record <category>.<name> (  
    <type field-name>  
    ...  
)
```

where:

<category>

one of the element categories discussed above (component, view, bus, or connection).

<name>

provides a name for the element type. Each element category is a separate namespace. For instance, it is legal to define both a `view.ethernet` and a `bus.ethernet`. The maximum length of *name* is 64 bytes.

<type field-name>

defines one or more fields that define the properties of the element. The maximum length of <type field-name> is 64 bytes. Certain fields must be defined, while other fields have particular functions—these are described in the following section.

- All element type definitions must have a field called `Name`, which names the element instance.
- Connections must have two additional fields, `Object1` and `Object2` that name the endpoints of the connection.
- `Glyph_State` allows you to enable or disable glyph state propagation to the element.
- Components may be SNMP devices. The following fields provide SNMP information that is included with requests to the SNMP proxy agent (see Chapter 19, “SNMP Support,” for more information):
 - `SNMP_RdCommunity` specifies the SNMP community name to use when reading attribute values.
 - `SNMP_WrCommunity` specifies the SNMP community name to use when writing attribute values.

- `SNMP_Vendor_Proxy` specifies a non-SNM SNMP proxy agent. This should only be specified when a vendor has supplied an SNMP proxy agent that communicates with the SunNet Manager SNMP proxy agent.
- `SNMP_Timeout` specifies the number of seconds that the SNMP proxy agent is to wait for a response to requests sent to the target element.
- For `bus.ethernet` elements, `Default_Proxy` specifies the default proxy system for any elements that are discovered in the subnet. When the Discover Tool is run for the subnet, the name specified in this field is the default proxy system for each element in the subnet.
- `Label` is an optional field you can add that allows a label that is different from `Name` to appear under the glyph that corresponds to an element instance. To use the `Label` field, enter in a string in the Element Properties sheet. If no value is entered, the value in the `Name` field is displayed. If a single hyphen (-) is entered in the `Label` field, neither the `Label` nor the `Name` value is displayed under the element's glyph.

Other fields are optional and provide information used when displaying the element's properties.

The following rules are used to display an element's properties:

- The properties are the list of field names and values in an element's record.
- Any field whose name begins with underscore is not displayed in an element's properties.
- Embedded underscores in field names are converted to blanks in the Properties list. For instance, the field `User_Name` will be displayed in the Properties under *<User Name>*.

The following example defines some element types:

```
record component.sun4 (
    string[64]      Name
    string[40]      IP_Address
    string[40]      User
    string[40]      Location
    string[80]      Description
    string[40]      SNMP_RdCommunity
    string[40]      SNMP_WrCommunity
    string[40]      SNMP_Vendor_Proxy
    int            SNMP_Timeout
    string[256]     SNMP_SysObject_ID
    string[40]      Physical_Address
)

record view.subnet(
    string[64]      Name
    string[80]      Description
    string [40]     NetMask
    enum stateProp Glyph_State
)

record connection.rs232 (
    string[64]      Name
    string[64]      Object1
    string[64]      Object2
    string[80]      Description
)

record bus.ethernet (
    string[64]      Name
    string[40]      IP_Network_Number
    string[80]      Description
    enum stateProp Glyph_State
    string[64]      Default_Proxy
)
```

18.1.1 Glyph Definition (for Component or View Element Type)

Elements that are either a component or view are displayed as a glyph. An `elementGlyph` instance in the element's schema specifies the glyph for a component or a view.

An `elementGlyph` instance has the following format:

```
instance elementGlyph (  
    ( <element> "<icon-path-name>" )  
    ...  
)
```

Note that the `elementGlyph` instance must be placed *after* the element type component record. If the `<icon-path-name>` begins with a slash (/), it is treated as an absolute path. Otherwise, the path name is relative to the directories specified by the Icon Directories setting in the Console Properties Locations category. See Chapter 14, "Console," for more information about Console Properties settings.

The following example shows an `elementGlyph` instance that specifies several glyph/component mappings:

```
instance elementGlyph (  
    ( component.sun3                sun3.icon)  
    ( component.sun4                sun4.icon)  
    ( component.sun386              sun386.icon)  
    ( component.laserwriter         laserwriter.icon)  
    ( component.pc                  pc.icon)  
    ( component.genhost             mainframe.icon)  
    ( component.genws               generic-ws.icon)  
    ( component.router              router.icon)  
    ( component.bridge              bridge.icon)  
    ( component.lanbox              lanbox.icon)  
)
```

Note that you can have up to 1024 icons or raster files defined in the runtime database; this includes background images as well as element glyphs.

18.1.2 User Command Definition for an Element Type

A UNIX command may be defined to appear in the Tools menu of a component, view, bus, or connection by including an `elementCommand` instance in the element schema. This provides the ability to quickly execute any UNIX command by pointing to a glyph and selecting the command. The `elementCommand` instance has the following syntax:

```
instance elementCommand (
    ( <element> "<menu-name>" "<cmd-string>" )
    ...
)
```

where:

<element>

specifies an element type. All elements of this type will have the command in their glyph menu. The maximum length of **<element>** is 64 bytes.

<menu-name>

specifies the name of the command to appear in the menu. The maximum length of **<menu-name>** is 32 bytes.

<cmd-string>

specifies the UNIX command string to execute. Any word in the command string of the format `%field` will be replaced by the value of `field` for that element. The maximum length of **<cmd-string>** is 1024 bytes.

Note that the `elementCommand` instance must be placed *after* the element type component record.

The following example defines two commands that can be run from elements of type `component.sun4`.

```
instance elementCommand (
    (component.sun4 "Rlogin..." "$SNMHOME/bin/snm_cmdtool
$SNMHOME/bin/snm_exec rlogin %Name")
    (component.sun4 "Telnet..." "$SNMHOME/bin/snm_cmdtool
$SNMHOME/bin/snm_exec telnet %Name")
)
```


As shown in the example above, the environment variables `$SNMHOME` or `$SNMDBDIR` can be used, even if you have not explicitly set them.

If you have not set `SNMHOME` or `SNMDBDIR`, then

- `/usr/snm` and `/var/adm/snm` are assumed for Solaris 1.x environments
- `/opt/SUNWconn/snm` and `/var/opt/SUNWconn/snm` are assumed for Solaris 2.x environments.

Thus, for example, if SNM is installed on a SunOS 5.x machine, the string `"$SNMHOME/bin/snm_cmdtool"` is expanded to `"/opt/SUNWconn/snm/bin/snm_cmdtool"` if `SNMHOME` is not otherwise defined.

18.2 Element Instance Definition

Element and request instances are defined using cluster records. A cluster record provides a means for collecting multiple records to make up a single instance. A cluster record contains an initial record defining the instance's type followed by one or more records providing additional information about the element or request.

For example, the following cluster record defines a component `.sun4`:

```
cluster(  
    component.sun4 ( SunEng14 1.2.2.388 Engineer23  
                    "Building 2, Room 146"  
                    "Sun 4/110 - monochrome,  
                    diskfull" )  
    membership ( Subnet2 100 230 0 )  
    agent ( etherif )  
    agent ( hostif )  
    agent ( hostmem )  
    proxy ( hostperf SunEngServer )  
    agent ( layers )  
    proxy ( ping SunEngServer )  
    agent ( rpcnfs )  
    glyphColor ( 220 220 250 )  
    connect ( SunLAN )  
)
```

The first record in the example, `component . sun4`, defines the instance's type. It must be a record defined in your element's schema file. The first field in the record is normally the name of the element, in this case, `SunEng14`. The string `"Building 2, Room 146"` is a single field with a maximum length of 64 bytes. When a field contains an ASCII string with embedded blanks, surround the string with double quotes.

Note – Element names are not case-sensitive. `SunEng14` is the same as `suneng14`.

The other fields in the first record correspond to the `component . sun4` record definition in the elements schema. See the section on “Element Type Definition” for more information. Except for when the instance is a connection, these other fields are optional. Connections have two additional required fields that specify the endpoints of the connection.

The second record in the example, the `membership` record, places the instance in a view. Views represent a logical grouping of elements. A view may contain other views. The way you choose to arrange instances in views is up to you. Some examples are:

- Lay out your network elements in a “logical” (network) view as well as in a “physical” (machines in a building) view. For instance, display all the routers in the `Routers` view and also place them in views corresponding to their location within the network.
- Arrange views in a circular (or loop) arrangement where a view that is a member of the `Home` view can also contain the `Home` view.

The first field of the `membership` record indicates the name of the view—in this case, `Subnet2`. A special view, `Home`, refers to the view that is displayed when the Console is started. This is also the default view if no `membership` records are specified.

The remaining fields in the `membership` record provide positioning coordinates for the instance and are optional. For bus and connection elements, five coordinates are specified: X coordinate, Y coordinate, Z value (used when glyphs are stacked on top of each other in a view), and X and Y coordinates of the second point of the element. For other types of elements, only the X, Y, and Z coordinates are specified. If the coordinates are not specified, the instance will be automatically positioned in the view by the Console.

Although the example shows only a single `membership` record in the cluster, an instance may be placed into several views by including multiple `membership` records in the cluster.

The third through ninth records in the example—the `agent` and `proxy` records—provide the list of available agents and proxies for the instance. The first field in the record provides the name of the agent or proxy. `proxy` records contain a second field that supplies the name of the system providing the proxy.

The tenth record in the example is a `glyphColor` record. This record provides the color for the glyph. It is optional and, if omitted, the glyph is transparent. The three fields provide the red, green, and blue intensities, respectively. Each intensity can vary between 0 and 255. Transparent is all zeros. White is all 255s.

The final record in the example is a `connect` record. It is optional and indicates that the element is connected to another instance. This will cause a line to be drawn to the connected instance if it appears in the view. `connect` records are valid only in an instance that is a component, bus, or view. The example connects `SunEng14` to `SunLAN`. Connections made via `connect` records are referred to as simple connections since they have no name.

Cluster records can also contain a `glyphState` record, which should not be modified. If you are creating a new cluster record, you do not need to define the `glyphState` record; glyph state is assumed to be normal.

18.3 *Connection Definition*

A connection is another element category supported by the SunNet Manager Console. Connections are like any other element and can be selected and named, and can contain subviews containing other elements. A connection is created by creating a cluster whose first record is of type `connection.<type>`. The endpoints of the connection are named by the second and third fields in the `connection.<type>` record that correspond to the fields `Object1` and `Object2`.

The following example shows a definition of a link type of connection:

```
cluster (
    connection.link ( ExtLink BigHost SomeNet )
    membership ( Home 400 320 32766 401 88 )
    glyphColor ( 255 0 255 )
)
```

In this case, `BigHost` and `SomeNet` are the elements that are the endpoints of the connection named `ExtLink`. The connection will only appear in the view `Home` if the elements `BigHost` and `SomeNet` both exist in that view.

18.4 Background Definition

Background images may be associated with a view instance by including a `viewBackground` instance record in an instance file. The image should be in raster file format. Icon format is supported if the file ends with the extension `.icon`. However, large backgrounds in icon format load much more slowly than raster format files.

Before you specify a background, you need to create the view instance that will be associated with the background. If the icon path name begins with a slash (`/`), it is treated as an absolute path. Otherwise, the path name is relative to the directories specified by the `Icon Directories` setting in the `Console Properties Locations` category. See Chapter 14, “Console,” for more information about `Console Properties` settings. The following example shows a definition of a background image file called `USmap.iml` with the `Home` view:

```
instance viewBackground (
    ( Home USmap.iml )
)
```

The background image file can be a maximum of 64 bytes. Note that the background definitions must be placed *after* the view instance definition. You can have up to 1024 icon or raster files defined in the runtime database; this includes background images as well as element glyphs.

Note – The preferred method of adding a background image to the current view is to invoke the **View**►**Add Background** option in the Console window. Refer to your *Site/SunNet/Domain Manager Application and Agent Development Guide* for more information.

18.5 Tools Menu Definition

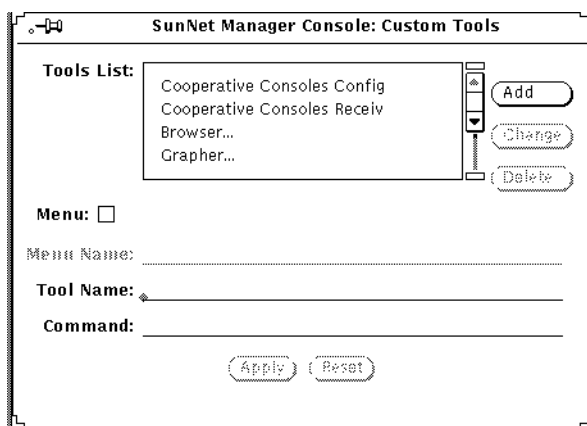
A UNIX command may be defined to appear in the Tools menu of the Console, allowing you to launch the command as a separate process. This definition is accomplished by an `elementCommand` instance, similar to the `elementCommand` definition for an element type.

Starting with version 2.3, a new database record is defined in the `snm.glue` file:

```
(SNM_Console_Menu "<cmd-name>"
```

This record will allow you to create submenus to facilitate execution of SunNet manager commands. The other way to create submenus is through the Console as follows:

1. Click **SELECT** on **Tools**►**Customize**►**Menu** to receive the following screen:



2. Enter a menu or tool name on the appropriate line.

3. Click SELECT on Apply to save the changes to your runtime database.

Each menu list on the Console provides the opportunity to add submenus, tools or a command menu item.

The following example defines tools that can be invoked from the Console's Tools menu.

```
instance elementCommand (
(SNM_Console "Browser..." "$SNMHOME/bin/snm_br")
(SNM_Console "Grapher..." "$SNMHOME/bin/snm_gr")
(SNM_Console "Snapshot.." "snapshot")
(SNM_Console_Menu "Discovery")
(SNM_Console "IP Discover..." "$SNMHOME/bin/snm_discover")
(SNM_Console "IPX Discover.." "$SNMHOME/bin/snm_ipx_discover")
}
(SNM_Console_Menu "cisco")
  (SNM_Console "Router Mgmt..." "router_mgmt")
  (SNM_Console "Switch Mgmt..." "switch_mgmt")
}
(SNM_Console_Menu "Performance")
```

As shown in the example above, any string in the form \$name is replaced by the expanded <name> environment variable. The string "\$SNMHOME/bin/snm_br" is expanded to "/usr/snm/bin/snm_br" if you have the SNMHOME environment variable set to /usr/snm. (If you have not set the SNMHOME environment variable, /usr/snm is assumed for Solaris 1.1 installations; /opt/SUNWconn/snm is the default for Solaris 2.x installations.)

18.6 Definition of Requests

This section discusses the format of requests. Most users will want to create requests from the Console rather than manually building requests as described here. However, this information is included as reference information on how requests are stored in the management database.

Every request, whether active or held, is defined by a cluster record. The definitions of the records used in these clusters can be found in the Console definitions file. This file is located at:

- /usr/snm/struct/snm.glue, on Solaris 1.1 installations
- /opt/SUNWconn/snm/struct/snm.glue, for Solaris 2.x installations.

Like other instances, the type of request (data or event) is determined by the first record in its cluster. Requests for data begin with a `dataRequest` record, while event requests begin with an `eventRequest` record.

18.6.1 Data Requests

The following example is a cluster record representing a Data Request:

```
cluster(
  dataRequest ( hostperf.data.0 0 SunEng14
                SunEngServer hostperf data
                ``Interval' `Count' Times"
                10 5 "" "" "" undefined True
                False "Save Request" undefined
                hostperf.data SunEng14 )
  dataAttribute ( cpu% True 1 True 1473696 None
                  0 )
  dataAttribute ( intr False 0 False 0
                  "Delta Values" 1515136 )
  rqstState ( Idle 0 0 )
  membership ( SunEng14 460 260 0 )
)
```

The first record is the `dataRequest` record. See Table 18-1.

Table 18-1 Data Request — `dataRequest` Record

Field #	Field Name	Description
1	Name	Name of the request. Displayed under the request's glyph and in messages about the request. Not sent to the agent. Maximum length is 64 bytes.
2	_serialnumber	Serial number of the request. Set serial number to 0 for new requests.
3	_targetsystem	Name of the target system. Passed to the agent as the <code><system></code> name. If the next field is NULL, this field is also used to determine the location of the agent. Maximum length is 64 bytes.
4	Proxy_System	Name of the proxy system. Used to determine the location of the agent. If null, the value of the previous field is used to determine the location of the agent. Maximum length is 64 bytes.

Table 18-1 Data Request — dataRequest Record

Field #	Field Name	Description
5	_agent	Name of the agent. Maximum length is 64 bytes.
6	_group	Name of the group/table. Maximum length is 64 bytes.
7	Interval	Interval field passed to agent.
8	Count	Count field passed to agent.
9	Key	Key field passed to agent. Maximum length is 128 bytes.
10	Restart	Boolean indicating whether to automatically restart the request if it dies or the system on which it is running dies. Value must be either True or False.
11	Defer_Reports	Boolean indicating whether sending responses should be deferred until requested by the Console. Value must be either True or False.
12	On_Completion	Disposition of request when it terminates. Value must be either Delete Request or Save Request .
13	Options	Options field passed to agent as an argument with the name of the string in netmgt_optstring. Maximum length is 128 bytes.
14	Log_to_File	File name for data reports. If NULL, reports are sent to the data log window. Maximum length is 128 bytes.
15	_fileport	Internal variable for the Console. Set to undefined for new requests. Do not modify for existing requests.
16	To_Program	Sends reports to a program. This is the name of the program.
17	_prmpport	Internal variable for the Console. Set to undefined for new requests. Do not modify for existing requests.
18	_timestamp	Internal variable for the Console. Set to undefined for new requests. Do not modify for existing requests.
19	_groupId	<Agent>.<group> name (should match the values of parameters 5 and 6).
20	_targetobject	Object associated with the request. Used to determine the name that should be put in error messages, which glyph should be blinked, etc. Must be the name of an element in the database. Maximum length is 64 bytes.
21	SnmpRedirect	Boolean indicating whether an SNMP request is to be redirected to another (vendor's) SNMP proxy agent.
22	Start_Date	Start time in the format mm/dd/yy
23	Stop_Date	Stop time in the format mm/dd/yy
24	Start_Time	Start time in the format hh:mm (hh 0-23, mm 0-59)
25	Stop_Time	Stop time in the format hh:mm (hh 0-23, mm 0-59)

Following the dataRequest record are zero or more dataAttribute records.

These records control how data is displayed by the Console. If no `dataAttribute` records are in a request element, all data is displayed in the Data Reports Log (or specified file). See Table 18-2 for a description of the `dataAttribute` record.

Table 18-2 Data Request–`dataAttribute` Record

Field #	Field Name	Description
1	Attribute	Name of the attribute. Maximum length is 64 bytes.
2	Data_Log	Boolean indicating whether attribute should be displayed in the Data Log. Value must be either True or False.
3	_asciiport	Internal variable for the Console. Set value to 0 for new requests. Do not modify for existing requests.
4	Indicator	Boolean indicating whether this attribute should be displayed in an indicator. Value must be either True or False.
5	_bargraphport	Internal variable for the Console. Set value to 0 for new requests. Do not modify for existing requests.
6	Strip_Chart	If and how the attribute should be displayed in a strip chart. Value must be either None, Absolute Values, or Delta Values. None specifies no strip chart, while Absolute Values and Delta Values specify a strip chart based on <i><absolute></i> or <i><delta></i> (current - previous) values.
7	_stripchartport	Internal variable for the Console. Set value to 0 for new requests. Do not modify for existing requests.
8	Graph_Tool	If and how the attribute should be displayed in the Grapher. Value must be either None, Absolute Values, or Delta Values. None specifies no graph, while Absolute Values and Delta Values specify a grapher based on <i><absolute></i> or <i><delta></i> (current - previous) values.
9	_grapherhandle	Internal variable for the Console. Set value to 0 for new requests. Do not modify for existing requests.

Following the `dataAttribute` records is the `rqstState` record. See Table 18-3 for a description of the `rqstState` record.

Table 18-3 Data Request — `rqstState` Record

Field #	Field Name	Description
1	<code>state</code>	State of the request. If you wish the request to be automatically activated when the Console is started, set this field to <code>Active</code> and set the <code>Restart</code> field in the <code>dataRequest</code> record to <code>True</code> . Otherwise set this field to <code>Idle</code> . Refer to the <code>rqstmgr_state</code> enumeration in <code>snm.glue</code> for other possible values.
2	<code>reqflags</code>	Internal variable. Always set this to 0.
3	<code>timestamp</code>	Internal variable. Always set this to 0.

The final record is a `membership` record. This record indicates in which element's subview the request glyph will appear. See Table 18-4 for a description of the `membership` record.

Table 18-4 Data Request — `membership` Record

Field #	Field Name	Description
1	<code>view</code>	Name of the element in whose subview the request glyph appears. Normally, this would be the same as the final value of the <code>dataRequest</code> record. Maximum length is 64 bytes.

The remaining fields in this record specify X and Y positioning. These fields should be left blank and the Console used to position the request glyph.

18.6.2 Event Requests

The following shows a cluster record representing an Event Request:

```
cluster(
  eventRequest ( hostperf.data.1 1 SunEng14
                 SunEngServer hostperf data Once
                 0 1 " " " False False False
                 "Delete Request" undefined
                 hostperf.data SunEng14 )
  eventAttribute ( cpu% "Greater Than" 80
                  "Threshold Not Set" " "
                  "Blink Glyph" " " Low )
  rqstState ( Idle 0 0 )
  glyphState ( 32 )
  membership ( SunEng14 )
)
```

The first record is an `eventRequest` record. See Table 18-5 for a description of the `eventRequest` record.

Table 18-5 Event Request — `eventRequest` Record

Field #	Field Name	Description
1	Name	Name of the request. Displayed under the request glyph and in messages about the request. Not sent to the agent. Maximum length is 64 bytes.
2	_serialnumber	Serial number of the request. Internal serial number used by the tool. Set this field to 0 for new requests.
3	_targetsystem	Name of the target system. Passed to the agent as the <system> name. If the next field is NULL, this field is also used to determine the location of the agent. Maximum length is 64 bytes.
4	Proxy_System	Name of the proxy system. Used to determine the location of the agent. If NULL, the value of the previous field is used to determine the location of the agent. Maximum length is 64 bytes.
5	_agent	Name of the agent. Maximum length is 64 bytes.
6	_group	Name of the group or table. Maximum length is 64 bytes.
7	Interval	Interval field passed to agent.
8	Count	Count field passed to agent.
9	Key	Key field passed to agent. Maximum length is 128 bytes.

Table 18-5 Event Request — eventRequest Record

Field #	Field Name	Description
10	Restart	Boolean indicating whether to automatically restart the request if it dies or the system on which it is running dies. Value must be either True or False.
11	Send_Once	Boolean indicating whether the request terminates after the first report. Value must be either True or False.
12	Defer_Reports	Boolean indicating whether sending responses should be deferred until requested by the Console. Value must be either True or False.
13	On_Completion	Disposition of the request when it terminates. Value must be either Delete Request or Save Request.
14	Options	Options field passed to agent. Maximum length is 128 bytes.
15	_timestamp	Internal variable for the Console. Set to undefined for new requests. Do not modify for existing requests.
16	_groupId	<Agent>.<group> name (should match values of parameters 5 and 6).
17	_targetobject	Object associated with the request. Used to determine the name that should be put in error messages, which glyph should be blinked, etc. Must be the name of an element in the database. Maximum length is 64 bytes.
18	SnmpRedirect	Boolean indicating whether an SNMP request is to be redirected to another (vendor's) SNMP proxy agent.
19	Start_Date	Start time in the format mm/dd/yy
20	Stop_Date	Stop time in the format mm/dd/yy
21	Start_Time	Start time in the format hh:mm (hh 0-23, mm 0-59)
22	Stop_Time	Stop time in the format hh:mm (hh 0-23, mm 0-59)

Following the eventRequest record are one or more eventAttribute records. These records set thresholds that determine when an event should be reported. Each record can set up to two thresholds for a particular attribute. See Table 18-6 for a description of the eventAttribute record.

Note – Only one record can be defined for a particular attribute. If you define multiple `eventAttribute` records for a request, each `eventAttribute` record must be for a different attribute.

Table 18-6 Event Request — `eventAttribute` Record

Field #	Field Name	Description
1	Attribute	Name of the attribute. Maximum length is 64 bytes.
2	Relation1	Relation for first threshold. Refer to the enumeration <code>threshold</code> in <code>snm.glue</code> for a list of possible relations.
3	Threshold1	Value for the first threshold. In the example, an event report is generated if <code>cpu%</code> was greater than 80. Maximum length is 16 bytes.
4	Relation2	Relation for the second threshold. <code>Threshold Not Set</code> indicates there is no second relation.
5	Threshold2	Value for the second threshold. Maximum length is 16 byte.
6	Priority	Defines the priority of the event. Must be either Low, Medium, or High.
7	Glyph_Effect	Action when event is reported. Refer to the enumeration <code>visualopt</code> in <code>snm.glue</code> for a list of possible actions.
8	Audio_Effect	Action when event is reported. Refer to the enumeration <code>audioopt</code> in <code>snm.glue</code> for a list of possible actions.
9	Audio_File	Name of the audio file. Maximum length is 128 bytes.
10	Mail_To	Mail address. Maximum length is 128 bytes.
11	To_Program	Shell script or program. Maximum length is 128 bytes.
12	_relop	Internal value. Leave blank or set to <code>undefined</code> .
13	Stop Request	Boolean indicating to stop the request in case of failure.
14	Start Request	Name of request to be started when this request is stopped.

Following the `eventAttribute` records is a `rqstState` record. See Table 18-7 for a description of the `rqstState` record.

Table 18-7 Event Request — `rqstState` Record

Field #	Field Name	Description
1	<code>state</code>	State of the request. Set this field to <code>Active</code> to automatically activate the request when the Console is started. Maximum length is 64 bytes.
2	<code>timeout</code>	Internal value. Always set this to zero.
3	<code>reqflags</code>	Internal value. Always set this to zero.

The final record is a `membership` record. This record indicates in which element's subview the request glyph will appear. See Table 18-8 for a description of the `membership` record.

Table 18-8 Event Request — `membership` Record

Field #	Field Name	Description
1	<code>view</code>	Name of the element in whose subview the request glyph appears. Normally, this would be the same as the final value of the <code>dataRequest</code> record.

18.7 Duplicate Databases

Previously, when you loaded ASCII topology databases and duplicate entries were found, SunNet Manager would abort the load operation. Starting with the current version, you have the following options when duplicate entries are found:

- Abort.
- Ignore duplicates and proceed with the load.
- Replace the old element with the new one from the ASCII database.

The `snm.conf` configuration file contains the keyword **snm.load-mdb-duplicate-objects** to which you may add the appropriate value as follows:

Abort - database loading is aborted when a duplicate entry is found

Ignore - all duplicate entries are ignored and placed in a list in the error window as warnings

Replace - all duplicate entries are replaced by objects from the ASCII database. All duplicates are listed in the error window as warnings.

If the keyword is missing from the `snm.conf` file or an invalid value is specified, the Ignore value, which is the default, is implemented.

This chapter discusses the following topics:

- SNMP proxy agent operation
- Schema files
- SNMP host files
- Asynchronous event reports (traps)
- SNMP object IDs mapped to strings
- SNMP Version 2 support

Use the SNMP proxy agent to get data and event information from and set attribute values for devices that are managed through SNMP.

Note – SunNet Manager also provides an SNMP agent for Sun workstations called the `snmpd` daemon. The SunNet Manager Console communicates with the `snmpd` daemon through the SNMP proxy agent. The `snmpd` daemon also allows Sun workstations to be managed by other SNMP management stations. For more information about the `snmpd` daemon, see the `snmpd(8)` man page.

This chapter discusses the operation of the SunNet Manager SNMP proxy agent and trap daemon. Using the `build_oid` utility to map SNMP object identifiers to strings is also described. For step-by-step information on how to add SNMP devices to your management database and set up the SNMP proxy agent and trap daemon, refer to “Part 1: Network Management Tasks.”

Note – This chapter assumes that you are familiar with SNMP concepts. If you will be creating your own enterprise-specific schemas, you should also understand SNMP group, table, and attribute definitions.

19.1 *SNMP Proxy Agent Operation*

The proxy agent runs on Sun workstations—either the one on which the management application is running or another workstation on the network. SunNet Manager management applications, such as the Console, communicate with the SNMP proxy agent (`na.snmp`) using the same RPC-based protocol as with other SunNet Manager agents. The SNMP proxy agent communicates with other devices using the SNMP protocol defined in RFC 1157, as shown in Figure 19-1.

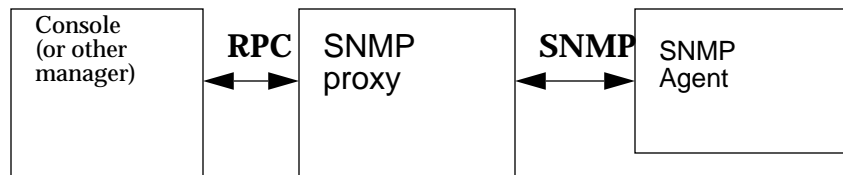


Figure 19-1 SNMP Proxy Agent

The proxy agent allows you to manage any number of management information bases (MIBs) in which you can define either standard SNMP MIB objects or enterprise-specific objects. The proxy agent uses a schema file to map objects described in a MIB and SunNet Manager attributes. A schema file is the representation of a MIB used by SunNet Manager. How to generate schema files from MIBs is discussed later in this chapter. To ensure successful operation, a schema file must have a set of object definitions that are identical to those in the MIB—see Figure 19-2.

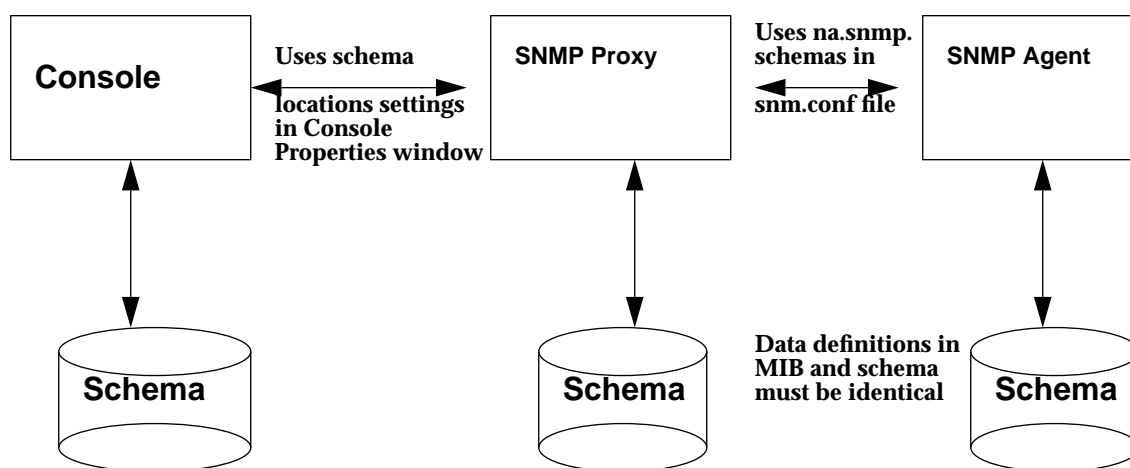


Figure 19-2 MIB and Schema Definition

Several SNMP schemas are supplied with SunNet Manager:

- `snmp.schema` describes MIB I, as defined by RFC 1156.
- `snmp-mibII.schema` describes MIB II, as defined by RFC 1213.
- `snmpv2-mibII.schema` describes MIB II, as used by SNMP version 2. See the “SNMP Version 2 Support” section for a description of SNMPv2 support in the current release of SNM.

`sun-snmp.schema` describes the MIB associated with the SNMP agent (`snmpd`) for Sun workstations. This schema file provides MIB II support with Sun enterprise-specific extensions. For more information about the schemas for various RFC MIBs, refer to the `snmpd(8)` manual page.

Except for the two MIB II files (which differ only in the RPC number specified), each of the schema files listed above is a subset of the file that follows it. That is, `snmp.schema` is a subset of the two MIB II files, which are, in turn, are a subset of `sun-snmp.schema`

The SNMP proxy agent can simultaneously access any of the above-mentioned schemas, as well as other enterprise-specific schemas that you might create. Both the Console and the SNMP proxy agent need to access the same schema information. If the SNMP proxy agent and the Console are running on different systems, identical schema files must be accessible to both systems. The Console uses the Schema Locations setting of the Console Properties Locations category to locate the directories where the schema files reside. SNMP schema files on the Console are treated like any other schema files; they define groups, tables, attributes, etc., for the user interface. The SNMP proxy agent uses the keyword `na.snmp.schemas` in the `snm.conf` file to locate the directories where the SNMP schema files reside.

The following section describes in detail how the SNMP proxy agent works. Note that many of the operations of the proxy agent are defined by arguments passed in the request or with keywords in the `snm.conf` file on the proxy system. You should refer to the `snm.conf(5)` manual pages for the keywords that are related to the SNMP proxy agent.

When the SNMP proxy agent starts up (normally via `inetd`) it loads all the SNMP schemas located in the directories specified by the keyword `na.snmp.schemas` in the `snm.conf` file on its host system. Only SNMP-related schemas (schemas that contain an `rpcid` keyword value of '100122') are loaded.

When the SunNet Manager SNMP proxy agent receives a request for an SNMP agent on a particular device, it performs the following sequence of operations:

1. It checks whether there are any new or modified SNMP related schema files since the last request. If the proxy agent finds a new or modified schemas in any of the directories specified by the `na.snmp.schemas` keyword in the `snm.conf` file on the proxy's system, it loads the schema file.
2. It passes the request to an existing agent subprocess or forks a new subprocess, if needed, to handle the request. A single subprocess can handle multiple SNMP requests from an instance of a management application. The maximum number of subprocesses that the SNMP proxy agent can fork is set by the keyword `na.snmp.max-subprocs` in the `snm.conf` file. At

installation, this value is set to 20. The maximum number of requests that a subprocess can handle is set by the keyword `na.snmp.max-requests` in the `snm.conf` file. At installation, this value is set to 50.

3. It checks whether the request contained any optional arguments. Requests sent by the SNM management applications may include arguments in an SNMP request. For example, the Console sends SNMP-related arguments that are defined in the target element's Properties window. (Refer to the "Properties" section for information about the element's Properties window.) These arguments can include:
 - a. The name of the schema to be used with the request. If, for some reason, the specified schema does not contain the attribute group specified in the request, the proxy agent attempts to use the schema specified by the keyword `na.snmp.default-schema` in the `snm.conf` file on its host system. At installation, the default schema is set to be:
 - `/usr/snm/agents/snmp-mibII.schema` for Solaris 1.x installations
 - `/opt/SUNWconn/snm/agent/snmp-mibII.schema` for Solaris 2.x installations.This schema supports the MIB II definition.
 - b. A community name that specifies the SNMP community name the proxy agent is to use when reading or writing attribute values. If no community name is specified, `public` is used for both Get and Set requests.
 - c. A request timeout that specifies the number of seconds the proxy agent is to wait for a response to a request sent to the target system. If no request timeout is specified, the proxy agent uses the value specified by the keyword `na.snmp.request_timeout` in the `snm.conf` file on its system. At installation, the value is set to 5 (seconds).
4. This step only applies if the proxy agent receives a request from a management application that is not an SNM version 2.3 or later management application. If you are using the proxy agent for the current product with the Console, skip to Step 5.

If no schema name is specified in the request, the proxy agent searches various schemas for the attribute group specified in the request. Schemas are searched in the following order:

- a. The schema(s) specified for the target device in the SNMP host file on the proxy system. The SNMP host file also specifies the community names and request timeout to be used for the request.

Versions of the SNMP proxy agent prior to SunNet Manager 2.3 *required* that you define an SNMP host file so that the proxy agent could determine the correct schema file to use for each request. When used with the current version of the Console or other management application, the SNMP proxy agent does not *require* the SNMP host file to determine which schema file to use for each request. However, an SNMP host file will need to be defined in the following cases:

1. If you want to define the handling of traps on a per-device basis.
2. If the management application you are using with the current version SNMP proxy does *not* supply the schema name in a request.

The location of the SNMP host file is specified by the keyword `na.snmp.hostfile` in the `snm.conf` file on the proxy system. See the `na.snmp.hostfile(5)` man page for more information about the contents of the SNMP host file.

- b. The schema specified by the keyword `na.snmp.default-schema`. If this schema is used, community names are assumed to be `public` and the request timeout is assumed to be 5.
5. The proxy agent then sends an SNMP message to the device and waits for a response.

If the proxy agent is sending a Get request, the proxy sends up to three SNMP requests per reporting interval. (The maximum number of SNMP requests sent is specified by the keyword `na.snmp.max_attempts` in the `snm.conf` file—by default the value is set to 3.) For each PDU sent, the proxy waits for the specified request timeout for a response from the device. As mentioned previously, the request timeout can be an optional argument in the request. If it is not specified in the request, request timeout is either the request timeout value specified in the SNMP host file for the device or the value of the keyword `na.snmp.request_timeout` in the `snm.conf` file.

If the proxy agent does not receive a response after sending three SNMP requests, it sends a “No response from system” report to the Console. (The keyword `na.snmp.trap-if-no-response` in the `snm.conf` on the proxy system determines whether the proxy agent sends a trap or an error report. At installation, the keyword’s value is `true`—send a trap report.) The proxy agent then waits until the next reporting interval to send out another set of SNMP requests. If no reporting interval has been specified in the request, the proxy agent sends out SNMP requests every 30 seconds. If the proxy

agent does not receive a response when the last report is due, it sends both an error report and a trap report to the Console if `na.snmp.trap-if-no-response` is `true`.

If the proxy agent is sending a Set request, the proxy waits for the specified request timeout for a response before timing out. There is no attempt to re-send the request. The reason for this is as follows: Because UDP is the transport mechanism, there is no guarantee of message delivery, thus there is no way to determine whether the request or the response to the request was lost. If you do not receive a response from your initial Set request, you should perform a Get request to see whether or not the Set operation was successful.

6. When the proxy agent receives a response from the target device, it sends a report to the SNM rendezvous.

If the proxy agent does not receive an acknowledgment from the rendezvous within a specified time, the proxy agent terminates the request. The specified time that the proxy waits for the rendezvous to acknowledge the report is specified by the `na.snmp.report_timeout` keyword in the `snm.conf` file. At installation, the keyword's value is set to 5 (seconds).

Normally, if the SNMP proxy agent is not performing any requests, it will exit. The keyword `na.snmp.exit-if-no-requests` in the `snm.conf` file allows you to specify otherwise.

Asynchronous or unexpected reports (traps) from SNMP agents are handled by the trap daemon `na.snmp-trap`, which may run on one or more machines on the network. The daemon listens for incoming traps on the SNMP trap port and translates them to SunNet Manager traps. The trap daemon uses an SNMP trap file, which contains information on enterprise-specific traps and attribution of events to pseudo-devices. Refer to the section on "SNMP Host Files" for more information about SNMP traps.

19.1.1 Trap Filtering

Setting Host Specific Trap Filters

Starting with version 2.3, you can specify filters based on certain hosts. What you specify overrides any enterprise-specific filter. The keyword, `<host>`, can be used in the `snm.conf` file followed by the `ip_address` or host name and a

priority keyword. Each line beginning with the keyword <host> can be followed by subsequent lines describing the action, description, or priority for each trap.

Precedence Values

The `snm.conf` file specifies general priority for all traps. The trap daemon compares oid entries with specific settings specified in `snmp.traps` when it receives a trap. If a particular trap has an oid entry in `snmp.traps`, the `na.snmp-trap` daemon will take priority. The `na.snmp-trap` daemon searches for the hostname of the target trap. If the name is present, the daemon will use it as its priority.

19.2 Schema Files

If you do not already have a schema file for the device you want to manage, use the SNM `mib2schema` utility to convert an existing MIB file for the device.

Note – Nested groups or nested tables are *not* supported in SNM schema files.

You may need to manually edit the resulting schema file produced by `mib2schema`. The areas that are likely to require changes are:

- When `mib2schema` encounters an OCTET STRING, it inserts `-C ???` in place of a format string. If you want to format octet strings in a particular way, search the schema file for occurrences of `-C ???` to replace `???` with the required format string. If a format string is specified, the SNMP proxy agent formats each octet of the attribute value it receives from an SNMP agent before sending the attribute value to a SunNet Manager rendezvous. You may, however, choose *not* to enter any format string. In this case, the contents of the OCTET STRING will be printed as is.

The format string is the same as the `sprintf(3S)` format argument. Up to 16 octets can be formatted; each byte is sent to `sprintf` as a separate, unsigned character. For example, the format string:

```
%02.2X:%02.2X:%02.2X:%02.2X:%02.2X:%02.2X
```

causes an OCTET STRING containing a 48-bit Ethernet address to be formatted in standard colon notation (for example, `08:00:20:07:8F:93`).

Note – The format string and the length of the OCTET STRING to be formatted must match. All bytes specified in the format string are displayed. If the OCTET STRING is smaller than the format string, unexpected characters may be displayed in the formatted output.

Note that the `-C` format parameter is only used if the parameter `-T STRING` is specified for the attribute. If the parameter `-T STRING` is specified and `-C` format is not specified, the attribute is displayed as either octets or as a string, depending upon whether the attribute is an octet or display string.

An example of the characteristics string for the `ifPhysAddress` attribute in the `ifStatus` table is shown below:

```
"-N ifPhysAddress -O 1.3.6.1.2.1.2.1.6 -T STRING -A RO
-C %2.2X:%2.2X:%2.2X:%2.2X:%2.2X:%2.2X -X equal -F 0"
```

This results in the display:

```
ifPhysAddress=08:00:20:09:A0:D5
```

- Some SNMP devices cannot return groups or tables with a large number of attributes; this is due to local space limitations. When this happens, the SNMP proxy agent returns an error message that the response is “too big”. This means that very large groups or tables need to be split into smaller groups or tables to be received by the SNMP proxy. `mib2schema` does not automatically split groups or tables. Generally, if a group has more than 15 fields, it is a good idea to split the fields up into smaller groups. You can choose your own name for subgroups.

In addition to the schema file, the `mib2schema` utility produces an object identifier file (with the `.oid` suffix) that contains a table of object identifiers and names. Run the `build_oid` utility to add the contents of the resulting object identifier file to the SNM Object Identifier Database. See the `build_oid(1)` manual page for more information.

`mib2schema` may also produce a trap definition file (with the `.traps` suffix), depending upon whether traps were specified in the MIB. Refer to (refer to Section 19.4, “Asynchronous Event Reports (Traps),” on page 19-12 for more information about using the trap definition file.

If `mib2schema` cannot determine the key for a table characteristics field in the schema file, it inserts `-K ???` into the schema file.

19.3 SNMP Host Files

The SNMP host file maps a request with a schema file. It also maps devices with trap files (refer to Section 19.4, “Asynchronous Event Reports (Traps),” on page 19-12. Create an SNMP host file *only* for the following conditions:

- You want to specify the handling of traps on a per-device basis.
- You are using the current version of the SNMP proxy agent with an SNM 1.x version management application.
- You are using a 1.x version of the SNMP proxy agent.

Note – If you do not meet any of the above conditions, you do not need to create an SNMP host file.

The name of the SNMP host file is defined by the `na.snmp.hostfile` keyword in the `snm.conf` file. The default location for the hosts file is:

- `/var/adm/snm/snm.hosts` for Solaris 1.x environments
- `/var/opt/SUNWconn/snm/snm.hosts` for Solaris 2.x environments.

Each line in the SNMP host file specifies information for a single device. The line contains five to seven fields, defined below (the first five fields are mandatory):

<code><host-name></code>	<code><read-community></code>	<code><write-community></code>	<code><request-timeout></code>	<code><schema-file></code>
<code><trap-file></code>	<code><vendor-proxy></code>			

where:

`<host-name>`

is the name of the SNMP device. This name must match the element instance name (system name) of the device in a Console view. The name is not case sensitive.

<read-community>

is the SNMP community name the proxy agent uses when reading attribute values from *<host-name>* and when sending traps.

<write-community>

is the SNMP community name the proxy agent uses when writing attribute values to *<host-name>*.

<request-timeout>

is the number of seconds the proxy agent waits for an SNMP response from *<host-name>*.

<schema-file>

is a colon-separated list of the names of the agent schema files that contain the group and attribute definitions for *<host-name>*. Each name must be an absolute path name that begins with a 'slash' (/). Note that each schema file must reside in a directory that is specified by the keyword `na.snmp.schemas` in the `snm.conf` file on the proxy system.

<trap-file> (optional)

is the name of the trap file that contains the trap definitions for *<host-name>*. This must be either an absolute path name that begins with a slash (/) or "-" to specify the default trap file. Note that the directory in which the trap file resides must be a directory that is specified by the keyword `na.snmp.schemas` in the `snm.conf` file on the proxy system. Note that the Console displays the entries in the most recently submitted trap file. If the SNMP host file points to two different trap files, data in the most recent one will be read. Refer to Section 19.4, "Asynchronous Event Reports (Traps)," on page 19-12 for more information about traps and the trap file.

<vendor-proxy> (optional)

is an optional field that specifies the name of a proxy system. If this field is set, the SNMP request is not sent to *<host-name>*, but will be sent instead to *<vendor-proxy>*. This parameter should be specified only when a vendor has supplied an SNMP proxy agent to manage a particular device or set of devices. In this situation, the vendor's SNMP proxy agent communicates with the SunNet Manager SNMP proxy agent via SNMP, but communicates with the target device using either SNMP or a *different* protocol. If you specify this field, you must specify a value for *<trap-file>*.

An example of an SNMP host file with three device entries is shown below:

```
sunbox public private 5 /opt/SUNWconn/snm/agents/sun-snm.schema
device1 admin admin 10 /snm/vendor/device1.schema /snm/vendor/device1trap.schema
device2 public administrator 10 /snm/vendor/device2.schema - proximo
```

19.4 Asynchronous Event Reports (Traps)

SNMP agents may return unsolicited or unexpected reports called traps. SNMP defines six generic trap types for the following conditions:

0	coldStart
1	warmStart
2	linkDown
3	linkUp
4	authenticationFailure
5	egpNeighborLoss

In addition, trap type 6 is used for enterprise-specific traps. Enterprise-specific traps contain an enterprise object identifier (OID) and an enterprise-specific trap number.

19.4.1 Trap Daemon Operation

The SNMP trap daemon (`na.snm-trap`) is installed with SunNet Manager agents and daemons, normally in the `agents` directory. The trap daemon translates received SNMP traps into SunNet Manager traps and forwards them to the Event Dispatcher on one or more management stations.

When the SunNet Manager SNMP trap daemon receives a trap, it performs the following sequence of operations:

1. It first determines what to do with the received trap. The SNMP trap daemon uses an SNMP trap file to carry out the following tasks:
 - Translate trap type numbers to ASCII strings
 - Determine if a trap should be discarded
 - Determine the database element (glyph) to which a trap should be attributed (in cases where elements represent pseudo-devices — components that do not have their own IP address but share the network address of a device they are parts of, or attached to)

Refer to Section 19.4.2, “SNMP Trap File,” on page 19-14 for more information about creating trap files.

To locate the appropriate trap file, the trap daemon first checks the SNMP host file to see if there is an entry for the device. The SNMP host file allows you to map specific devices to trap files. Refer to Section 19.3, “SNMP Host Files,” on page 19-10 for more information about the SNMP host file.

If the trap daemon cannot find a trap file for the device in the SNMP host file, it then searches a default trap file. The default trap file is specified by the `na.snmp-trap.default-trapfile` keyword in the `snm.conf` file on the system where the trap daemon is running. Normally, the file name is:

- `/var/adm/snm/snm.traps` for Solaris 1.x installations
- `/var/opt/SUNWconn/snm/snm.traps` for Solaris 2.x installations.

The default trap file allows you to specify the handling of traps on an enterprise or host basis.

If the trap daemon cannot find an entry for a trap type in a trap file defined in the SNMP host file *or* in the default trap file, it forwards the trap to the Event Dispatcher on the management station(s) using the precedence values described in “Precedence Values” on page 19-8.

If the trap daemon encounters any kind of syntax error while reading a trap file, it sends an error message indicating the line of the error and forwards the trap.

2. If the trap is not discarded, the trap daemon then searches through the loaded schema files for an OID match.

A trap may include a list of variables and their values that provide further information about the cause of the trap. Normally, the variable is in the form of a raw object identifier (OID). If the trap includes such variables, the SNMP trap daemon can translate the raw OIDs into ASCII text. If the variable is an enumerator data type, the trap daemon translates the variable’s integer value into an equivalent ASCII string. The translation for each variable can be specified in a SNMP schema file that contains the OID.

When the trap daemon is started up, it loads the following files:

- a. Schema files in the directories specified by the keyword `na.snmp.schemas` in the `snm.conf` file

- b. The schema file specified by the keyword `na.snmp.default-schema` in the `snm.conf` file. This file is normally `snmp-mibII.schema` in the `agents` directory.
- c. `snmp-mibII.schema` (in the `agents` directory), if it is *not* specified as the value of the keyword `na.snmp.default-schema`.

Note – The SNMP trap daemon does not load duplicate schema files. A particular OID can only be defined in one schema file.

If an OID match is found in a schema file, the trap daemon performs the translation before sending the trap to the Event Dispatcher. Since enumeration types for traps are not supported by SunNet Manager, the trap daemon translates enumerations to a string if it can find the enumeration definition in a schema file. Otherwise, the enumeration is returned as an integer. If the trap daemon is unable to find an OID match in the loaded schema files, the raw OID is forwarded to the Event Dispatcher.

Normally, if the trap daemon finds an OID match in a schema file, it only forwards translated attribute name/value pairs to the Event Dispatcher. The keyword `na.snmp-trap.raw` in the `snm.conf` file allows you to specify that the SNMP trap daemon return raw OIDs and their values without translation, in addition to the translated attribute names and values—see the `snm.conf(5)` manual page for more information.

3. The trap daemon then forwards the trap to the Event Dispatcher. By default, the trap daemon `na.snmp-trap` sends trap reports to the Event Dispatcher on the machine on which the trap daemon is installed. The keyword `na.snmp-trap.rendez` in the `snm.conf` file allows you to specify the names of one or more manager systems that should receive the traps processed by the SNMP trap daemon — see the `snm.conf(5)` man page for more information.

19.4.2 SNMP Trap File

As mentioned previously, the SNMP trap daemon uses a trap file to:

- Translate enterprise-specific trap type numbers to ASCII strings
- Determine trap priority
- Determine if a trap should be discarded

- Determine which database element (glyph) a trap should be attributed to (in cases where elements represent pseudo-devices — components that do not have their own IP address but share the network address of a device they are parts of, or attached to)

If you used the `mib2schema` utility to create your schema file and traps were specified in the MIB, `mib2schema` generates a trap file. Otherwise, you need to create a trap file for any of the above-mentioned purposes.

See Chapter 8, “Managing SNMP Devices” for information on setting trap priorities.

Note – If you use `glyph` entries in the trap file, the associated SNMP trap daemon must run on the host which has the SNM database where these elements reside. The trap daemon cannot be distributed to machines where the SNM Console is not installed because the trap daemon must access the SNM database to retrieve the properties of the specified elements.

19.4.2.1 Trap File Entries for Enterprise and Host-Specific Traps

The trap file may contain one or more lists of generic, enterprise-specific or host-specific traps. Each list begins with a line containing the keyword `enterprise` followed by the enterprise object identifier. Subsequent lines contain a trap number, trap name, and, optionally, the keyword `discard`. Comments may be added by beginning a line with a pound sign (`#`). The syntax for the trap file is shown below:

```
enterprise <enterprise-object-ID>
<trap-number>      <trap-name>      [discard]
```

<enterprise-object-ID>

is the enterprise object identifier. If you want to discard generic traps, specify zero for the *<enterprise-object-ID>*.

<trap-number>

is an integer that represents the trap number of an enterprise-specific trap. If you want to discard a generic trap, specify the trap type number: 0, 1, 2, 3, 4, or 5.

<trap-name>

is the ASCII string associated with the *<trap-number>*. *<trap-name>* can be up to 127 characters in length; blanks are not allowed.

The keyword `discard` specifies that the trap daemon should not forward the trap to the Event Dispatcher. As mentioned previously, traps are treated as high priority events. You can specify the `discard` keyword for traps that you do not wish to be notified about, such as system reboots.

For example, the first example below defines enterprise-specific traps for the Sun enterprise and specifies that generic trap type 0: is to be discarded. The second example defines host-specific traps for the host Odyssey.

```
# generic traps
enterprise 0
    0      cold-start      discard

# Sun Microsystems traps
enterprise 1.3.6.1.4.1.42
    1      CPU_Failure      high
    2      Power_Supply_Failure      low
    3      Network_Connection_Failure      medium
    4      Over_Heating      discard
    5      RealTimeClock_Failure      discard
```

```
#
#Sample traps
# host Odyssey
    1      CPU_Failure      high
    2      Power_Supply_Failure      medium
    3      Network_Connection_Failure      high
```

When searching the SNMP trap file for matches, the SNMP trap daemon only compares as much of the enterprise identifier as specified in the SNMP trap file. For example, a file with the line:

```
enterprise 1.3
```


would match *all* enterprises beginning with 1.3. Since the searching is performed from the top of the file, you should place more restrictive enterprise identifiers before less restrictive ones.

If you define a single trap file that applies to all or most of your SNMP devices, you should make this file the default trap file. For example, suppose that you do not want to see any generic traps for most of the SNMP devices in your management domain. You would define a trap file that specifies that generic trap types 0 through 5 be discarded and make this the default trap file. If there are a couple of critical devices for which you want to see generic trap types, define another trap file that does not specify that the generic trap types be discarded. Create an entry for each device in the SNMP host file and specify the second trap file. Because the trap daemon searches the SNMP host file first to match a trap to a device name, generic traps will be returned for only those devices specified in the SNMP host file.

You create the SNMP host file on the system where the trap daemon resides. Refer to “SNMP Host Files” for information on creating the host file. If you have only one trap file that applies to all your SNMP devices, you can specify the trap file as the default trap file—you would not need to create an SNMP host file.

19.4.2.2 *Attributing Traps to Glyphs*

Glyph entries in the trap file allow the user to attribute traps to database elements that represent pseudo-devices — network components that do not have their own IP address but share the network address of a device that they are components of, or to which they are attached.

An example would be glyphs that represent the ports on a hub. In this case the trap-attribution feature would allow the SNMP trap daemon to determine which port is the source of the event. When a trap is generated (for example, due to a hub port failure), the appropriate Console effect — such as blinking or color by priority — would then be applied to the glyph for that port rather than to the hub as a whole, facilitating fault-isolation.

If you have vendor software whose MIB supports this trap-distribution feature, there are five tasks that must be accomplished to enable it for your Console:

- The agent schema file that supports the trap-distribution feature — either provided by the vendor or created from the vendor MIB using the `mib2schema` utility — must be installed in the `agents` directory.

- An appropriate `glyph` entry must be added to an SNMP trap file on the system where the SNMP trap daemon runs (and this must be the system on which the SNM Console runs).
- The `snmp.hosts` file must also exist on the system where the trap daemon runs and must contain an entry that identifies the SNMP trap file to be used for interpreting traps from the originating device.
- An element definition must be provided for the pseudo-device in the `elements.schema` file (or a separate element schema file in the `agents` directory) and this definition must contain a field used to correlate this pseudo-device with a particular glyph. This field must match the appropriate field used by the enterprise-specific agent schema.
- An icon and iconmask to represent the pseudo-device must exist in one of the directories specified for icon locations in the Console properties sheet.

For information on creating an agent schema from a vendor MIB, refer to Section 19.2, “Schema Files,” on page 19-8.” How to point the SNM Console to the location of the icon and iconmask is discussed in “Locations” in Chapter 17, “Props Menu.”

The other tasks can be accomplished as follows:

1. Trap File Glyph Entries

The glyph entry in the trap file must begin with the keyword **glyph** and have the following syntax:

```
glyph <trap_attribute> <database_user_name> <component_type> <component_property>
```

<trap_attribute>

is the attribute whose value will be matched against the value of the property in the component property sheet to determine the glyph to be associated with the trap.

<database_user_name>

is the name of the runtime SNM database without the “db.” prefix. The user name must be provided because the trap daemon could have been started by the `inetd` process; in that case, the user owning the trap daemon will not be the same as the user running the Console. This field is used to create the `SNM_NAME` environment variable required to access the SNM database.

<component_type>

is the type of the component whose property is to be compared (for example, `component.pseudo`).

<component_property>

is the property of the component whose value will be compared against the trap attribute value.

A sample glyph entry would be the following:

```
glyph    ifPortKey    alfred    component.hubport    PortKey
```

In this example, `ifPortKey` is in the variable binding list of the trap Protocol Data Unit (PDU). To determine if this trap should be attributed to a particular glyph, the trap daemon is to look at the value of the property `PortKey` for an element. The trap is to be attributed to that glyph if the glyph's `PortKey` value matches the value passed in the trap's `ifPortKey` variable. `alfred` is the name of the SNMP database user and `component.hubport` is the element type.

2. SNMP Host File Entry

An entry must be added to the SNMP hosts file only if a schema file other than the trap daemon's default SNMP schema file is to be used to enable the trap-distribution feature. The hosts file resides in the same directory as the trap file. The syntax of host file entries is as follows:

```
<host-name>    <read-community>    <write-community>    <request-timeout>    <schema-file>  
<trap-file>    <vendor-proxy>
```

For other types of host file entry, *<host-name>* is the name of the SNMP device. However, for the glyph-distribution feature, *<host-name>* is to be the name of the host where the proxy agent resides. In a standard host file entry, the proxy system name is specified in the *<vendor-proxy>* field. However, for purposes of the pseudo-device trap-attribution feature, the optional *<vendor-proxy>* field is never used since this information is specified in the *<host-*

name> field. For more information about the SNMP host file, refer to Section 19.3, “SNMP Host Files,” on page 19-10.” An example of a host file entry to implement the trap-distribution feature would be the following:

```
bigguy    public private    20 /opt/SUNWconn/snm/agents/snmp-
pseudodev.schema /var/opt/SUNWconn/snm/snmp.traps
```

In this example, “bigguy” is the name of the host where the SNMP proxy agent resides. /opt/SUNWconn/snm/agents/snmp-pseudodev.schema is the vendor agent schema; the only reason the host file is necessary is to point to this file.

3. Adding an Element Definition for the Pseudo-device

You will need to add a component definition for the glyph that represents the pseudo-device. The component definition is created in the form of a record definition as in the following example:

```
record component.hubport (                                # generic sun 4
    string[64]      Name
    string[40]      User
    string[40]      Location
    string[80]      Description
    netaddress      Port_Key
    string[40]      SNMP_RdCommunity
    string[40]      SNMP_WrCommunity
    string[64]      SNMP_Vendor_Proxy
    int             SNMP_Timeout
)

# Glyphs for components

instance elementGlyph(
    ( component.hubport                                hubport.icon)
)
```

The fields in this element definition will be displayed in the properties sheet for glyphs of this type, as shown in Step 19-3.

udmpk16c-76 (component.sun-workstation)		
Name: udmpk16c-76		
IP Address: 129.146.76.30		
User: Room 232		
Location: port 1 on hub		
Description: FDDI Adaptor Card		
SNMP RdCommunity: public		
SNMP WrCommunity: private		
iostat		I/O statistics (SunOS 4.x)
iostat2		I/O statistics (Solaris 2.x)
ippath		IP path information
iproutes		IP route table and statistics
layers		protocol layer statistics
layers2		protocol layer statistics (for SunOS 4.x)
lpstat		line printer status and queue info
<input checked="" type="checkbox"/> ping	localhost	IP connectivity info
rpcnfs		RPC and NFS stats
sample		sample system info agent
Red:	200	<input type="text"/>
Green:	255	<input type="text"/>
Blue:	225	<input type="text"/>
<input type="button" value="Apply"/> <input type="button" value="Reset"/> <input type="button" value="Alias..."/>		
Browse		

Figure 19-3 Sample Properties Sheet for Pseudo-Devices

Note that underscores in the element definition are displayed as spaces in the field names. In this example, the Port Key field gives the attribute — a user-defined port number — that is used to match traps to glyphs. (This “Port Key” value is in the same format as an IP address though it is not an IP address.) For information on how to create element definitions, see the section on “Element Instance Definition.” This definition can be added to the `elements.schema` file or to another element schema file in the `agents` directory.

This component definition must include a field that will have the same value as the SNMP trap attribute that will be matched with it. The following restrictions apply:

- The trap attribute to be matched with the component property must be in the variable binding list in the trap PDU. This follows the regular members of a trap message, which are the following:

- Version
 - Community
 - Enterprise
 - Agent address
 - Generic trap type
 - Specific trap type for enterprise-specific trap
 - Time stamp
- For purposes of matching the element property to the SNMP attribute, only the SNMP types listed in Table 19-1 are supported.

Table 19-1 Supported SNMP and Schema Types

SNMP Type	SNMP Attribute Type	Element Schema Attribute Type
INTEGER	int	int
INTEGER	enum	enum
OCTET STRING	string	string
IPAddress	netaddress	netaddress
NetworkAddress	netaddress	netaddress
OBJECT IDENTIFIER	objectid	string

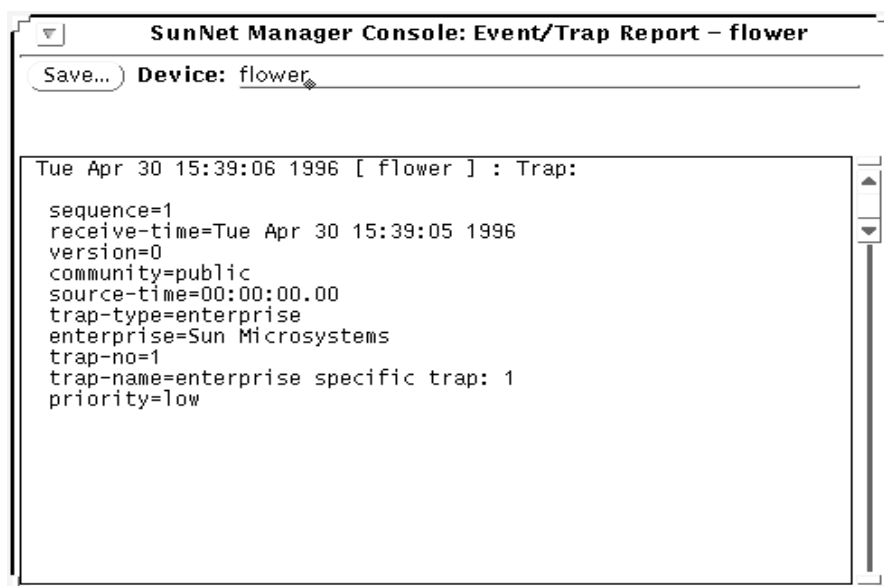
19.4.3 Trap Reports on the Console

On a Console system, trap reports are logged with other event and error reports and are displayed by selecting the Console’s View►Event/Trap Reports menu item.

Changes in a glyph’s state that results from a trap are propagated through the Console’s view hierarchy as if an event had occurred. You determine the priority of an SNMP trap as described in Chapter 8, “Managing SNMP Devices.” The type of signal that the Console uses to indicate received traps is specified in the Events and Traps category of the Console’s Properties window. By default, the Console blinks the glyph of the target element when a trap is received.

19.4.4 Interpreting a Trap Report

A typical trap report logged in the Event/Trap Reports window looks like the following:



```
SunNet Manager Console: Event/Trap Report - flower
Save... Device: flower
Tue Apr 30 15:39:06 1996 [ flower ] : Trap:
sequence=1
receive-time=Tue Apr 30 15:39:05 1996
version=0
community=public
source-time=00:00:00.00
trap-type=enterprise
enterprise=Sun Microsystems
trap-no=1
trap-name=enterprise specific trap: 1
priority=low
```

Figure 19-4 Trap Report

The lines in the trap report are described below:

The header is the line at the top which shows the date, time and hostname of the device which generated the trap.

sequence

a counter showing the sequence number of the trap; also keeps track of the number of traps generated since the device was last booted

receive-time

the time the Console received the trap

version

returned by the device which generated the trap

community

the SNMP trap community name sent by the device which generated the trap

enterprise

enterprise name of the device which generated the trap

source-time

length of time between when the device was booted and when the trap was generated

trap-type

whether this trap is generic or enterprise-specific

trap-no

where in the sequence of traps this one falls

trap-name

a name you have specified or a default name

priority

the priority of this trap (low, medium or high)

Note that underscores in the element definition are displayed as spaces in the field names. In this example, the Port Key field gives the attribute — a user-defined port number — that is used to match traps to glyphs. (This “Port Key” value is in the same format as an IP address though it is not an IP address.) For information on how to create element definitions, see the section on “Element Instance Definition.” This definition can be added to the `elements.schema` file or to another element schema file in the `agents` directory.

This component definition must include a field that will have the same value as the SNMP trap attribute that will be matched with it. The following restrictions apply:

- The trap attribute to be matched with the component property must be in the variable binding list in the trap PDU. This follows the regular members of a trap message, which are the following:
 - Version
 - Community

- Enterprise
 - Agent address
 - Generic trap type
 - Specific trap type for enterprise-specific trap
 - Time stamp
- For purposes of matching the element property to the SNMP attribute, only the SNMP types listed in Step 19-2 are supported.

Table 19-2 Supported SNMP and Schema Types

SNMP Type	SNMP Attribute Type	Element Schema Attribute Type
INTEGER	int	int
INTEGER	enum	enum
OCTET STRING	string	string
IPAddress	netaddress	netaddress
NetworkAddress	netaddress	netaddress
OBJECT IDENTIFIER	objectid	string

19.4.5 Trap Reports on the Console

On a Console system, trap reports are logged with other event and error reports and are displayed by selecting the Console's View►Event/Trap Reports menu item.

Changes in a glyph's state that results from a trap are propagated through the Console's view hierarchy as if an event had occurred. SNMP traps are treated as high-priority events. The type of signal that the Console uses to indicate received traps is specified in the Events and Traps category of the Console's Properties window. By default, the Console blinks the glyph of the target element when a trap is received.

19.4.5.1 Remaining Fields

Remaining fields, if any, are variables whose values are returned by the trap. Some traps do not return variables. These values are usually relevant to the reason the trap was generated. When the trap daemon receives a trap, it attempts to translate variables from OIDs into ASCII text before forwarding the trap. See the section on “Trap Daemon Operation” for more information.

19.4.6 Mapping SNMP Object IDs to Strings

One of the data types supported by SunNet Manager is an object identifier. Since object identifiers are numeric (for example, 1.2.3.4.5), a database is provided so the Console can display the object identifier using a more meaningful string. The Object Identifier Database (OID) provides the information for this mapping. The `build_oid` utility builds the OID. The Console uses the database to interpret object identifiers.

You will probably want to build the OID when new object identifiers are added to the database. Input files for `build_oid` are ASCII files with the suffix `.oid`. Each file has a descriptive name followed by its object identifier. If the descriptive name contains special characters, enclose it in double quotes. You can include comment lines by beginning them with a pound sign (`#`). When `mib2schema` is used to convert a MIB to a schema file, it also generates a `.oid` file.

To locate input files, `build_oid` uses the directories specified by `SNMHOME/agents` and `SNMHOME/schemas`. If you have not defined the `SNMHOME` environment variable, the directories are:

- `/usr/snm/agents` and `/usr/snm/struct` for Solaris 1.x installations
- `/opt/SUNWconn/snm/agents` and `/opt/SUNWconn/snm/struct` for Solaris 2.x installations

You can also specify other directories as input arguments (see the `build_oid(1)` man page for more information). SunNet Manager provides three input files which are normally found in the `/usr/snm/agents` directory:

- `enterprises.oid` contains the mappings for a variety of enterprise and organization identifiers.
- `snmp.oid` contains the mappings for object identifiers for MIB I of the SNMP protocol.

- `sun-snmp.oid` contains the mappings for object identifiers referenced by the Sun SNMP agent, `snmpd`. The Sun agent implements a superset of MIB II; thus, this file also serves as a `snmp-mibII.oid` file.

The environment variable `SNMDDIR` specifies the directory for writing the output database `oid.dbase`. If you have not specified the `SNMDDIR` environment variable, the directories are:

`/var/adm/snm` for Solaris 1.x installations

`/var/opt/SUNWconn/snm` for Solaris 2.x installations.

19.5 SNMP Version 2 Support

This section assumes you are familiar with SNMPv2 concepts. Instructions for installing and de-installing SNMPv2 are in your installation guide for Solaris 2.x/x86.

SunNet Manager provides a proxy agent that supports SNMPv2. This proxy agent allow you to get data and event information from and set attribute values for devices managed through SNMPv2.

There is also an SNMP agent for Sun workstations called the `snmpv2d` daemon. The Console communicates with this daemon through the SNMP proxy agent. The `snmpv2d` daemon also allows Sun workstations to be managed by other SNMPv2 and SNMP stations. For more information about the `snmpv2d` daemon, see the `snmpv2d(8)` man page.

The following sections discuss the differences between SNMP and SNMPv2. For information about the SNMPv2 configuration files, see the following man pages:

`v2install(1)`, `acl.pty(5)`, `agt.pty(5)`, `context.pty(5)`, `mgr.cnf(5)`, `mgr.pty(5)`, `snmpv2d.conf(5)`, and `view.pty(5)`.

Note – When the Discover tool locates SNMP devices on your network, it cannot determine whether the devices support functionality specific to SNMPv2.

19.5.1 SNMPv2 Enhancements

The key enhancements from SNMP to SNMPv2 are in the following categories:

- Structure of Management Information (SMI)
- Protocol operations
- Manager-to-manager capability
- Security

19.5.1.1 Structure of Management Information

The SMI for SNMPv2 is based on the SMI for SNMP. The SNMPv2 SMI provides more extensive specification and documentation of managed objects and MIBs.

Several new data types were created for SNMPv2. These include a 64 bit-counter (`Counter64`) and the `UInteger32` type which allows representation of integers in the range 0 to $2^{32} - 1$.

The SNMPv2 `OBJECT-TYPE` macro includes an optional `UNITS` clause, which contains a textual definition of the units associated with an object. This clause is useful for any object that represents a measurement in units (ex. "seconds"). The `OBJECT-TYPE` macro for SNMPv2 also includes a `MAX-ACCESS` clause which allows you to specify the maximum level of access.

19.5.1.2 Protocol Operations

SNMPv2 has three new protocol data units (PDU). The SNMPv2 trap PDU works in a way similar to that of the SNMP trap PDU, but it uses the same format as most other SNMPv2 PDUs. This eases the receiver processing task.

A major enhancement for SNMPv2 is the `GetBulkRequest` PDU. This PDU can significantly minimize the number of protocol exchanges required to retrieve a large amount of management information.

The third additional PDU is the `InformRequest` PDU. This is sent by an SNMPv2 manager, on behalf of an application, to another SNMPv2 manager. The PDU provides management information to an application using the second SNMPv2 manager.

19.5.1.3 *Manager-to-Manager Capability*

Manager-to-Manager operations are supported through the use of the manager-to-manager MIB. This MIB is a set of objects which describe the behavior of an SNMPv2 entity acting in a manager roll. For more information, see RFC 1451.

19.5.1.4 *Security*

SNMPv2 uses the Secure SNMP (S-SNMP) party concept for security. Improvements over S-SNMP include the elimination of ordered delivery mechanism and simplification of the clock synchronization algorithm. In addition, SNMPv2 introduces the context concept. Contexts provide for more efficient storage of access control and MIB view information. SNMPv2 uses both DES and MD5 for message security and authentication.

19.5.2 *SNMPv2 Files*

You can install SNMPv2 as an agent (`snmpv2d`), a manager (`na.snmpv2`), or both. The required files are installed as part of the current product. Installation steps are the same for both agents and managers. After the current SunNet Manager product is installed, create the three configuration files required by the `v2install` script. The files are:

- `agents` - contains names of hosts on which the `snmpv2d` agent will be installed
- `mgrs.v1` - contains names of hosts that will be running SNMPv1 managers (`na.snmp`)
- `mgrs.v2` - contains names of hosts that will be running SNMPv2 managers (`na.snmpv2`)

See the `v2install(1)` man page for detailed information about these files.

Procedures for installing (or removing) SNMPv2 software are in your installation guide for Solaris 2.x/x86.

19.5.3 *Using the v2mib2schema Program*

A program, `v2mib2schema`, has been included with the current product to allow you to translate your own SNMPv2 MIBs to SNM schema files.

Be aware that SunNet Manager schemas do not have the flexibility of SNMPv2 MIBs, so changes to the MIB may be necessary before `v2mib2schema` can successfully parse it.

Although `v2mib2schema` parses TEXTUAL-CONVENTIONS clauses, it currently ignores them, so later references to the new types will cause syntax errors. See the `v2mib2schema(5)` man page for more details.

This chapter discusses the following topics:

- Starting the Results Browser
- Loading files
- Report Streams
- Streams Menu
- Selecting Streams
- Folders
- Customizing the Browser

You can examine data stored in the following files:

- Console Event/Trap log file. This file is normally located at:
 - `/var/adm/snm/event.log` for Solaris 1.x installations
 - `/var/opt/SUNWconn/snm/event.log` for Solaris 2.x installations.
- Console Data Report log entries that have been saved to a disk file. (Refer to “Data Reports” for more information.)
- Data Report files. You use Save option in the Console Data Report window to create a log file of data reports.
- Other files that use the data format specified in `snm logfile(5)`.

Each instance of the Browser can only work with other SunNet Manager tools that have the same user name.

20.1 Starting the Browser

Invoke the Browser from either the Tools menu in the SunNet Manager Console or from a command line. It runs as a separate process from the Console and remains when you quit the Console. To start the Browser from the Console, press MENU on the Tools button and drag and release on the Browser menu option.

To invoke the Browser from the command line, use the following command:

```
mgrhost% <tools-path>/bin/snm_br [-b] [<filename1>] [<filename2>] ..
```

- *<tools-path>* is /usr/snm for Solaris 1.x installations.
- *<tools-path>* is /opt/SUNWconn/snm for Solaris 2.x installations.

When you start the Browser from a command line, you can optionally specify files to be automatically loaded into the Results Browser. File names can include wildcards. When invoked without any file names specified, the Results Browser displays the window shown in Figure 20-1.

If the Browser is started from a command line with the `-b` option, this causes graphs to be plotted on a white background, rather than the default black background, if the Grapher is invoked from the Browser.

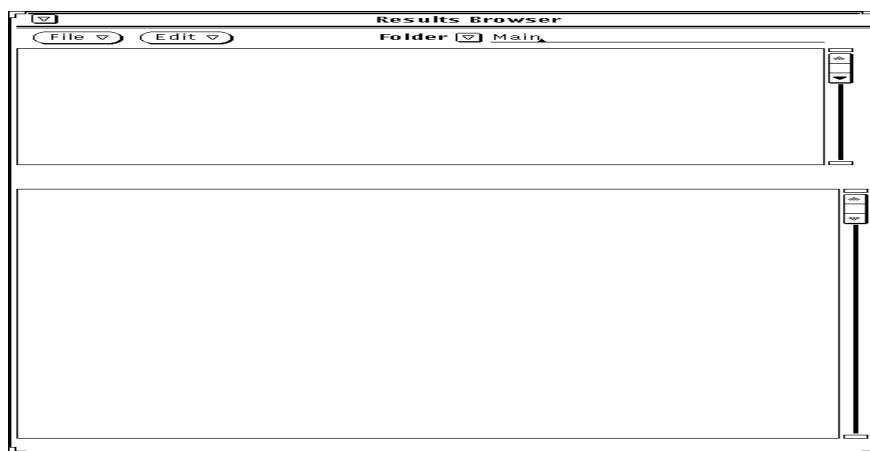


Figure 20-1 Results Browser Window

20.2 Loading Files

To load a file (which must be in the prescribed format) into the Browser, press MENU on the File button and release MENU on the Load option. A pop-up window appears, prompting you for the name of a file to be loaded. See the example in Figure 20-2, “Load Window.”

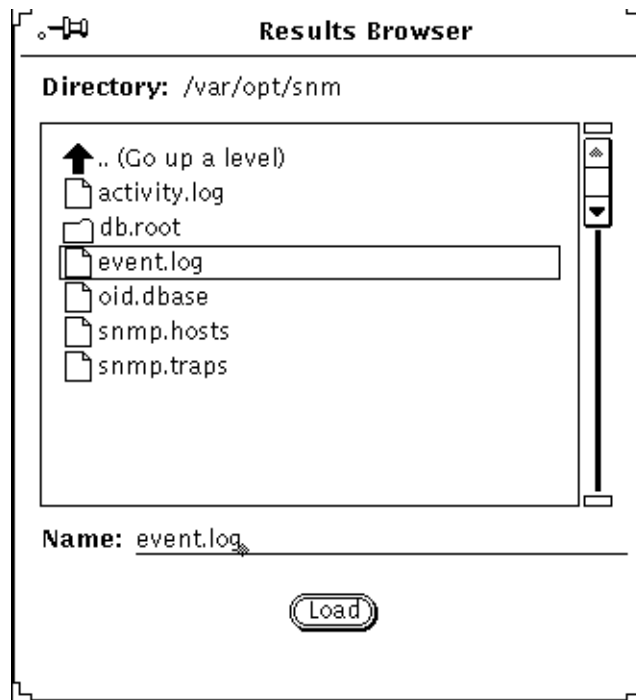


Figure 20-2 Load Window

To load a file, double-click SELECT on the file name or its icon or type in the file name at the Name prompt and then click SELECT on the Load button. You can load multiple files into the Browser, one at a time; however, you cannot specify wildcards in any file name.

Note – Because the Browser keeps open every file it reads, there is a limit to the number of files you can load in a Browser session. This limit is determined by the number of file descriptors allowed by the C-shell to a single process; the default is 64. You can increase this number up to 256 by using the `limit` command in the C-shell. You can also run multiple Browser sessions at the same time.

20.3 Report Streams

When you load a file into the Browser, the data in the file is automatically organized into report streams, which appear in the scrolling list in the upper pane of the Browser window. See Figure 20-3, “Results Browser Report Streams.”

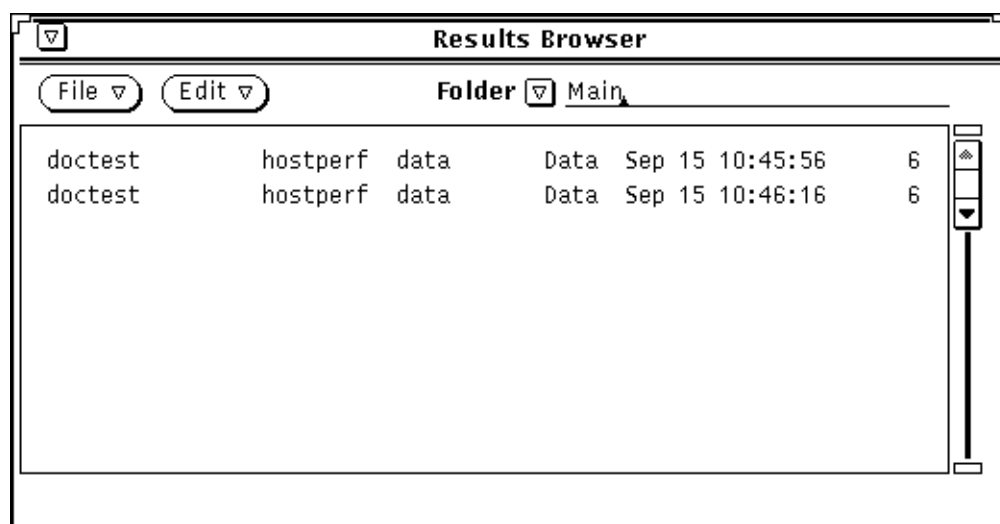


Figure 20-3 Results Browser Report Streams

Report streams are logical groupings of reports from agents. Each stream contains the results of a request from a particular requester address at a particular time. The Browser also distinguishes streams by report type (data, event, error, etc.). For example, if a file contains both data reports and error reports resulting from an individual request, the data reports are listed in a separate stream from the error reports.

Each report stream is defined in the following format in the upper portion of the Browser window:

<code><system></code>	<code><agent></code>	<code><group></code>	<code><type></code>	<code><date></code>	<code><number_of_reports></code>
-----------------------------	----------------------------	----------------------------	---------------------------	---------------------------	--

where:

`<system>` is the name (up to 15 characters) of the target system (the host where the managed object resides).

`<agent>` is the name of the agent.

`<group>` identifies the attribute group.

`<type>` identifies the type of report (data, event, trap, error).

`<date>` is the time stamp of the original request from the manager.

`<number_of_reports>` indicates the number of reports in the stream.

To view the contents (individual reports) of a particular stream, double-click SELECT on the stream you want to view. Individual agent reports from the selected stream are displayed in the lower portion of the Browser window, as shown in Figure 20-4, “Agent Reports from Selected Stream.”

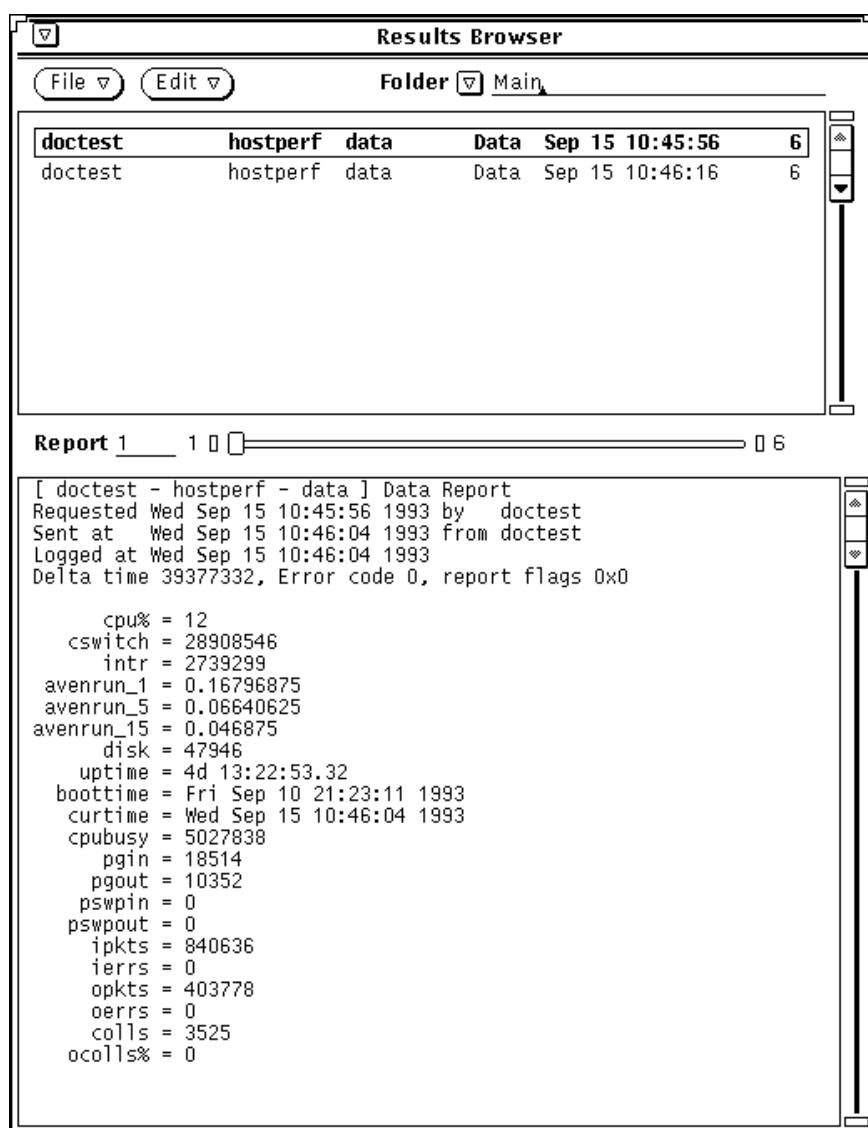


Figure 20-4 Agent Reports from Selected Stream

If more than one report exists in the selected stream, a slider bar appears. To view a specific report, type in the report number on the Report line to the left of the slider bar and press Return. You can also view reports by pressing

SELECT on the drag area (the small rectangular area between the ends of the slider bar) and dragging the mouse pointer to the right or the left until the desired report number is reached.

You can clone, delete, or print reports from the selected stream by pressing MENU in the Report window to bring up the Report menu. The Report menu is shown in Figure 20-5, "Report Menu."

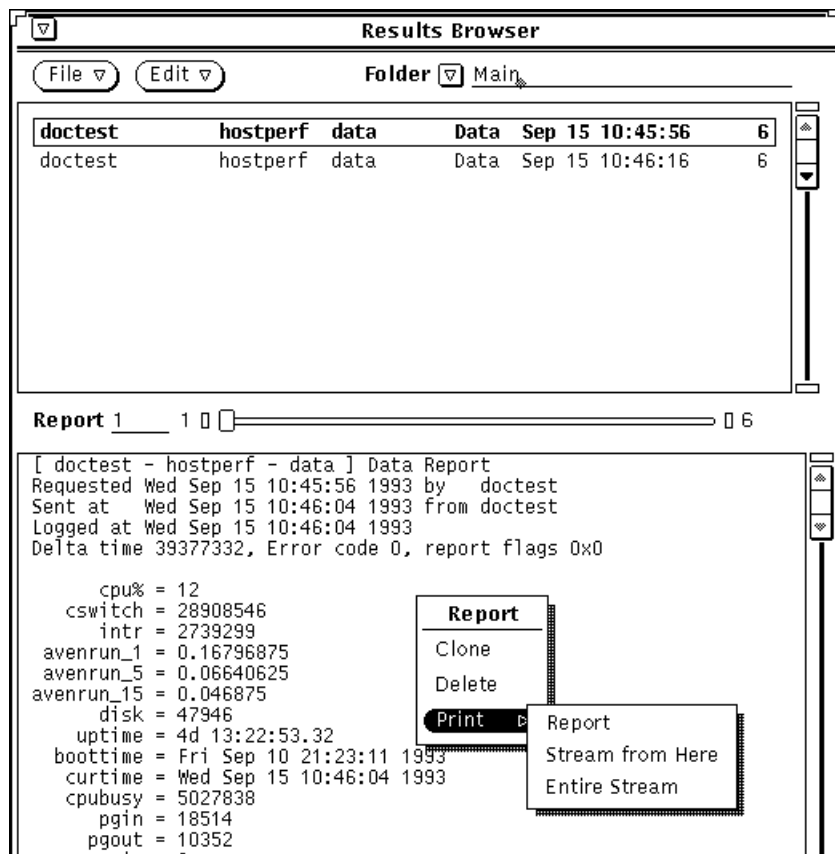


Figure 20-5 Report Menu

Report Menu functions are described as follows:

- To clone a report, select the Clone option. This causes a copy of the current report to be displayed in a pop-up window so that its contents can be compared with other reports. To make the pop-up window disappear, unpin it by clicking SELECT on the pushpin.
- To delete a report, select the Delete option.
- To print a report, press MENU in the Report window and move the pointer to one of the three Print Options:
 - Report—prints the currently displayed report.
 - Stream From Here—prints all reports in the stream, beginning with the currently displayed report.
 - Entire Stream—prints all reports in the stream.

By default, reports are printed to `lpr` on Solaris 1.1, or `lp` if SNM has been installed on a Solaris 2.x machine. You can choose a different printer by using the Tool Properties option of the Browser's Edit menu. Refer to the section on "Customizing the Browser" for more information.

If the reports to be printed exceed a set number of kilobytes, you are asked to confirm that you want the reports printed before printing commences. By default, this limit is set at 100 kilobytes. You can adjust this limit by using the Tools Properties option of the Browser's Edit menu. Refer to the section on "Customizing the Browser," for more information.

20.4 Streams Menu

To perform operations on one or more streams, invoke the pull-down Streams menu by pressing MENU in the top panel of the Browser window. You receive:

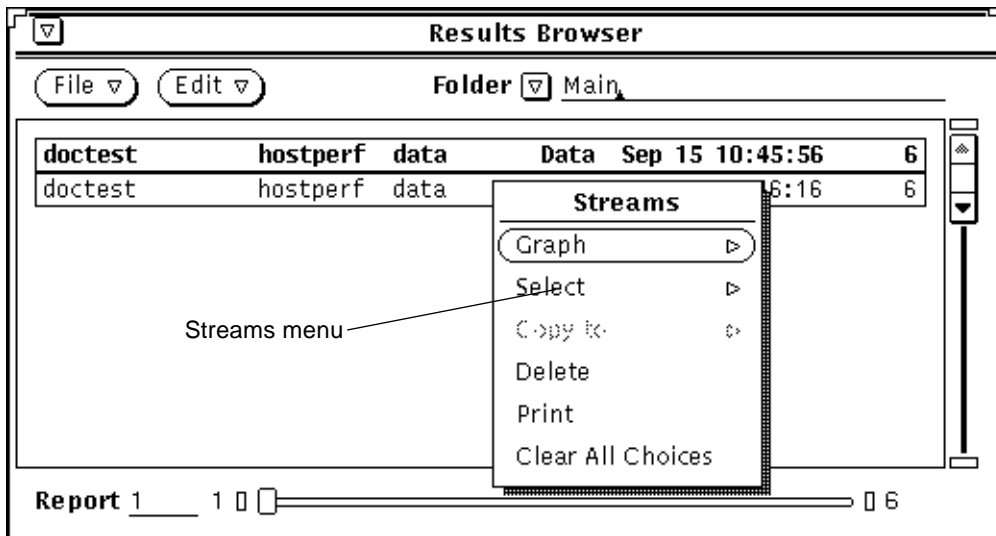


Figure 20-6 Results Browser Streams Menu

The following briefly describes Streams menu functions.

- Graph allows you to send selected streams to the Results Grapher. This function is described in detail in the section, "Sending Data to the Results Grapher."
- Select allows you to select streams based on certain properties. This function is described in detail in the section "Selecting Streams."
- Copy to copies selected streams to a folder. The use of folders is described later in Section 20.6, "Folders."
- Delete deletes selected streams.
- Print prints selected streams to your configured printer.
- Clear All Choices deselects any selected streams.

20.5 Selecting Streams

The Browser provides mechanisms for selecting streams based on certain user-specified properties. This function can be invoked by pressing MENU in the scrolling list and pulling right on the Select option of the Streams menu. You can select streams By System, By Agent.Group, or By Report Type. When you have made your selections, the names of the report streams in the scrolling list to which the specified properties apply are highlighted. Figure 20-7, “Streams Selection by System” illustrates selection by system.

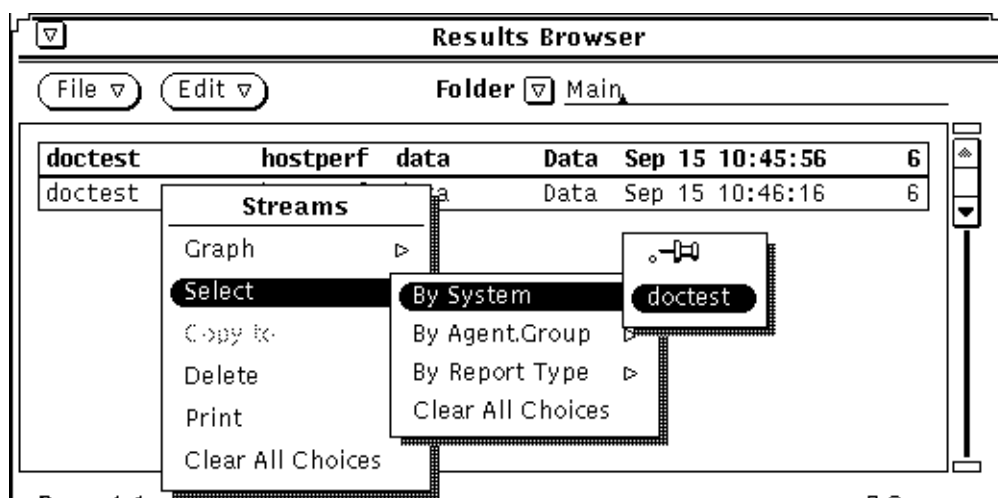


Figure 20-7 Streams Selection by System

20.5.1 Sending Data to the Results Grapher

You can have data in a report stream plotted using the Results Grapher. To send data to the Grapher, first select the desired streams. Press MENU in the scrolling list in the upper portion of the Browser window to invoke the Streams menu, pull right on Graph and release on one of the listed attribute names (shown in an alphabetized list). Figure 20-8, “Sending Data to the Grapher” illustrates how this is done:

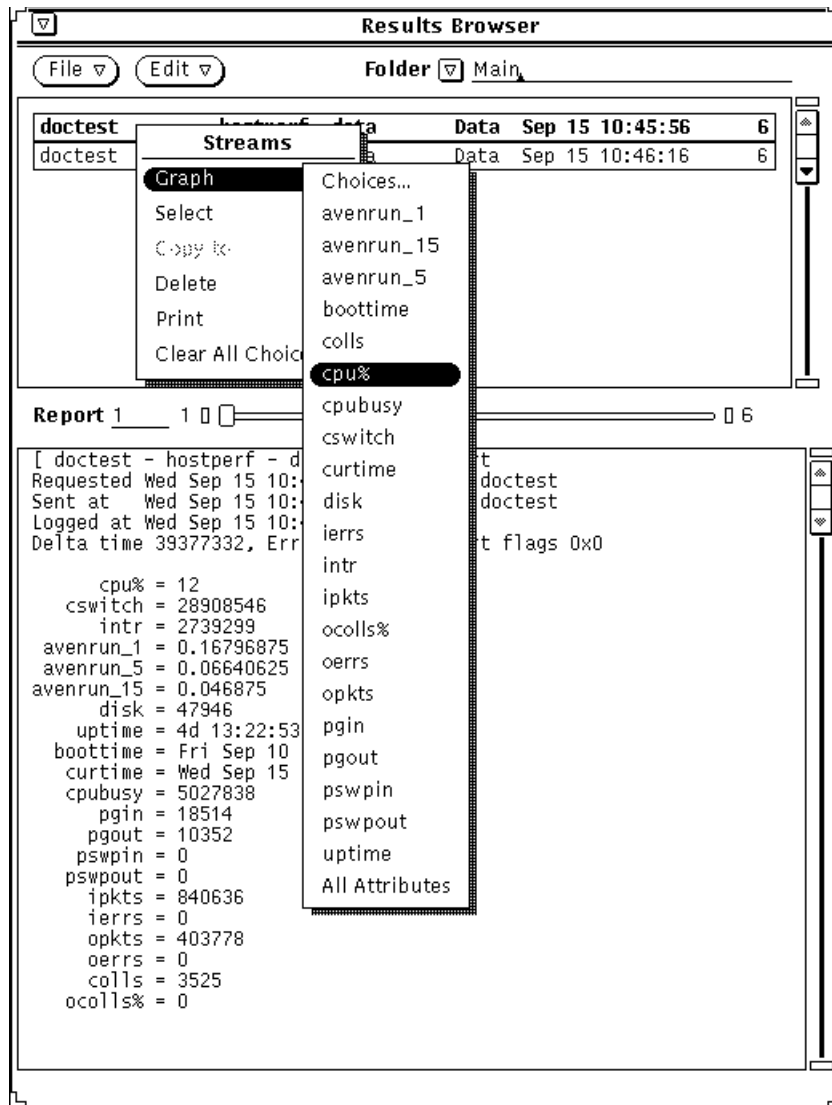


Figure 20-8 Sending Data to the Grapher

When you make this selection, the Browser sends a set of attribute and corresponding time values to the Results Grapher for plotting. Only integer, float, counter, gauge, timestamp, and UNIX time data may be plotted. Therefore, only the names of attributes of these data types are included in the menu list.

You can select more than one report stream and have each plotted on the same graph. For example, you can graph comparative data for multiple systems onto a single graph.

You can also graph multiple attributes from one or more selected report streams. To do this, pull right on either the All Attributes or the Choices option. The All Attributes option plots every integer and float attribute listed in the report stream. The Choices option displays a pop-up window with a list from which you can select the attributes to be graphed. Note that the All Attributes option is only available if there is more than one attribute that can be plotted.

If the Browser was started from a command line with the `-b` option, graphs will be plotted on a white background when the Results Grapher is invoked. Otherwise, if the `-b` option was not specified, the Results Grapher will display graphs on a black background

After information is sent to the Grapher, you can view the data in a Grapher session. See Chapter 21, “Results Grapher,” for information.

20.6 Folders

When you load a file into the Browser, the file is automatically loaded into the default folder `Main`. A folder is a *temporary* place to logically group report streams and store them for reference during a Browser session. You can create other folders to hold specified groups of report streams by using the Edit menu.

To create new folders, press MENU on Edit, drag the pointer to New Folder, and then release the MENU button. A folder named “New Folder” is created, which you can rename by typing the desired name on the line where “New Folder” is displayed, and pressing Return. The “Main” folder *cannot* be renamed.

To load files into a folder, press MENU on File and drag the pointer to the Load option. A pop-up window appears, prompting you for the name of the directory and file name of the file to be loaded. To load the file, click SELECT the Load button. You can load multiple files into a single folder, but duplicate reports are ignored.

You can also copy selected streams into a folder by using the Copy to option in the Streams menu. To do so, first select one or more streams by using the Select option of the Streams menu or by clicking SELECT on the desired stream in the scrolling list. Press MENU in the scrolling list and drag the pointer to Copy to. Continue to pull right to see a list of folder names; release the pointer on the desired folder. Figure 20-9, “Copying Streams to a Folder” illustrates copying streams to a folder.

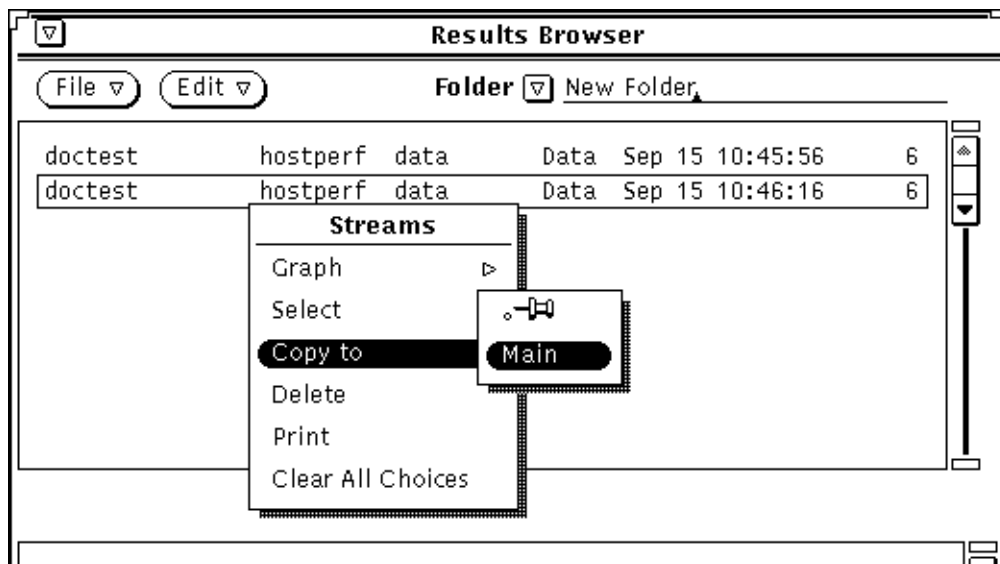


Figure 20-9 Copying Streams to a Folder

To empty a folder of all its streams, press MENU on Edit and drag the pointer to Empty Folder.

To delete a folder, press MENU on Edit and drag the pointer to Delete Folder. If you try to delete the Main folder, it will be emptied but not deleted.

As mentioned previously, folders are temporary places to keep related streams together. Once you exit the Browser, the folder names and their contents disappear. You can save the contents of a folder to a file for future access. To do so, press MENU on File and drag the pointer to the Save Folder option. In the pop-up window, specify a path and file name and click the Save button. In a future Browser session, you can load this file to view and manipulate the same set of streams.

Caution – Do *not* attempt to save the contents of a folder into a file that contains reports that are currently loaded into the Browser, as this will corrupt the file.

Because log files may be extremely large, individual reports are not kept in memory. Instead, pointers to the files are maintained. Therefore, if a file is overwritten while it is open, the file would be destroyed and the pointers to the file would be corrupted.

20.7 Customizing the Browser

You can customize the properties of the Browser by using the Tool Properties option of the Edit menu. Press the mouse MENU button on Edit and drag and release on the Tool Properties option. The Properties window appears, as in Figure 20-10, “Tool Properties Window.”

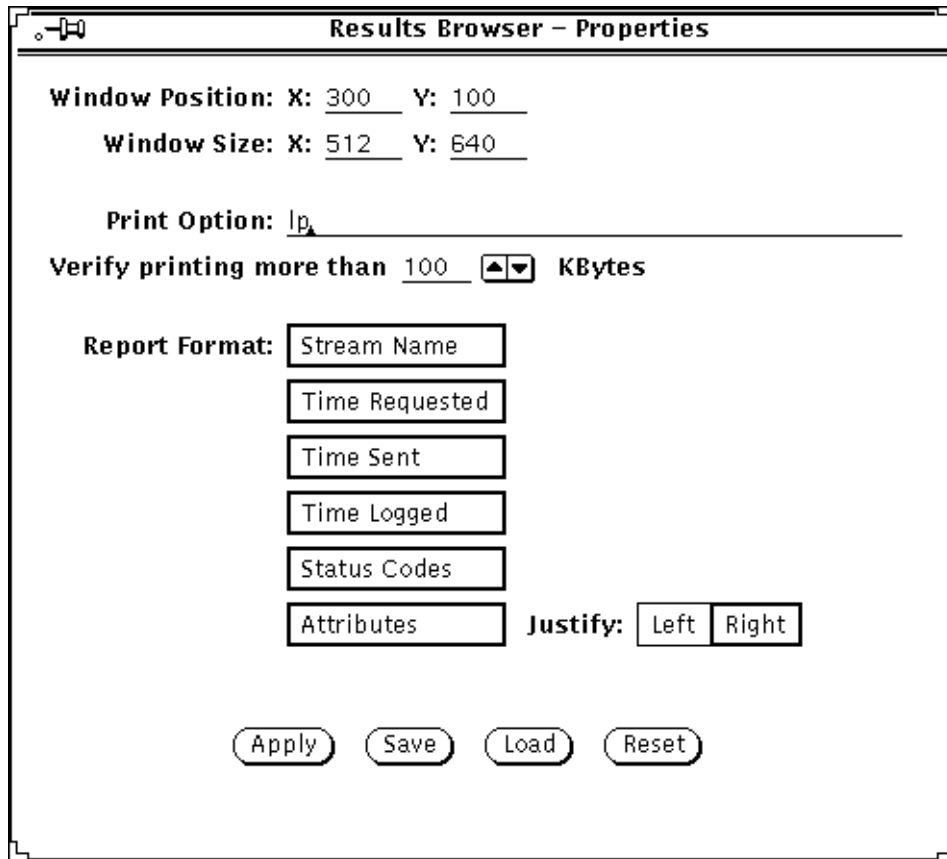


Figure 20-10 Tool Properties Window

The following fields in the Properties window affect the operation of the Browser. Any changes you make take effect when you press the Apply or Save buttons.

Window Position:

Defines the location (in pixels) of where the Browser window is displayed. Default values are 300 and 100 for x and y locations, respectively.

Window Size:

Defines the dimensions (in pixels) of the Browser window. Default values are 512 and 640 for x and y dimensions, respectively.

Print Option:

Specifies the printer name. `lpr` is the default printer for Solaris 1.1 installations. `lp` is the default printer for Solaris 2.x machines.

Verify printing more than (100) KBytes

Defines a print output size for printing report streams. For any output that exceeds the specified size, you will be asked to confirm whether you want to print the reports. Either enter in a new value on the line or click SELECT on the up- or down-arrow glyphs to increase or decrease the specified size.

Report Format:

Defines the appearance and contents of stream reports in the lower portion of the Browser window. All of the listed choices are selected by default, which causes all report information to be displayed in the report window. Click SELECT on any of the fields to deselect or select a choice. You can also choose to have attribute names displayed left- or right-justified on the screen. By default, attribute names are displayed right-justified.

The four buttons at the bottom of the Properties window operate as described below.

- **Apply** applies any changes that have been made to the fields in the Tool Properties window.
- **Save** saves window parameter changes to the `OpenWindows.xdefaults` file and applies any changes you made to the Report Format and Print Option parameters.
- **Load** loads window parameters from the X Resource Manager. You can then click the Apply button to apply these parameters to the Browser window.
- **Reset** sets the values in each of the fields to the last applied changes (the last time you clicked the Apply button) or to the values when the Properties window was first displayed.

Results Grapher



This chapter discusses the following topics:

- Starting the Results Grapher
- Results Grapher window
- Graph Properties window
- Displaying graphs
- Merging graphs

The Results Grapher is a graphing utility that you can use to visualize attribute values returned from Data requests or stored data sent from the Results Browser.

In a Data request, you can choose to have reported attribute values sent to the Results Grapher for display. (This is described in Chapter 15, “Requests Management.”) From the Results Browser, you can send data to the Results Grapher by selecting a report stream and pulling down a menu to graph selected attribute values from the report stream. (This is described in Chapter 20, “Browser.”)

Each instance of the Grapher can only work with other SNM tools that have the same user name.

21.0.1 Starting the Results Grapher

The Grapher can be invoked from either the Tools menu in the SunNet Manager Console or from a command line. Either way, the Results Grapher runs as a separate process from the SunNet Manager Console and remains running when you quit the Console. To start the Grapher from the Console, press MENU on the Tools button and drag and release on the Grapher menu option (see Figure 21-1).

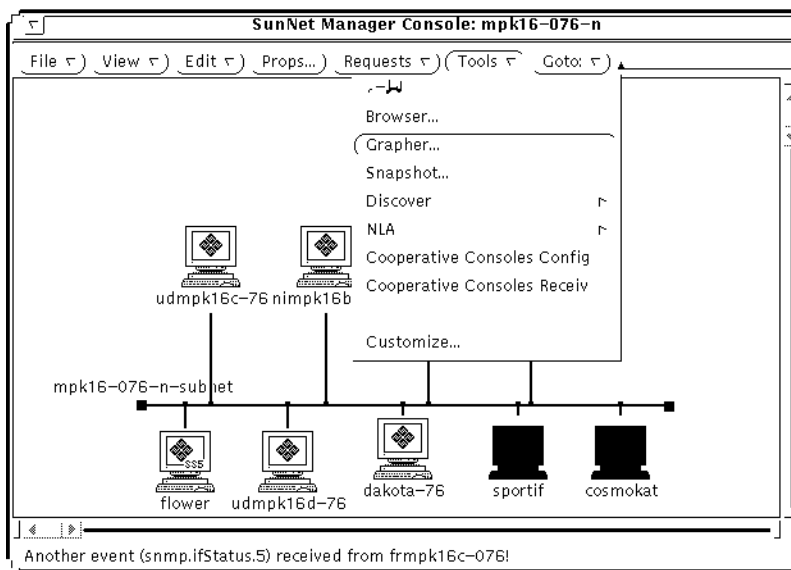


Figure 21-1 Invoking the Grapher from the Console

You can also invoke the Grapher from the command line by entering.

```
mgrhost% <tools-path>/snm_gr [-b]
```

where *<tools-path>* is:

- /usr/snm/bin for Solaris 1.x installations
- /opt/SUNWconn/snm/bin for Solaris 2.x installations.

If the `-b` option is not specified in the command line, graphs will be plotted on a black background. If the `-b` option is specified, graphs will be plotted on a white background. When you invoke the Grapher from the Tools menu, graphs are by default plotted on a black background.

To cause graphs to be displayed on a white background when you invoke the Grapher from the Tools menu, you must add the `-b` option to the `snm_gr elementCommand` instance in the `elements.schema` file. (Refer to Chapter 18, “Management Database,” for more information about modifying the `elements.schema` file.)

To cause graphs to be displayed on a white background when you invoke the Grapher from the Results Browser, start the Browser with the `-b` command-line option.

When the Grapher is invoked, the Results Grapher window appears. When you send data to the Grapher from the Browser, a graph window appears. The following describes these windows.

Note – When the Grapher starts, it writes its temporary RPC program number to `/tmp/snm_gr.rpcid.<user>` (`<user>` is either the name of the user or the value of the environment variable `SNM_USER`). Other applications, such as the Console, the Browser, or `snm_cmd`, use this file to send data to the Grapher.

21.1 Results Grapher Window

If you previously sent data to the Grapher, the names of the graphs appear in the scrolling list; if not, the list is empty. The Results Grapher window contains a scrolling list and four buttons, as shown in Figure 21-2.

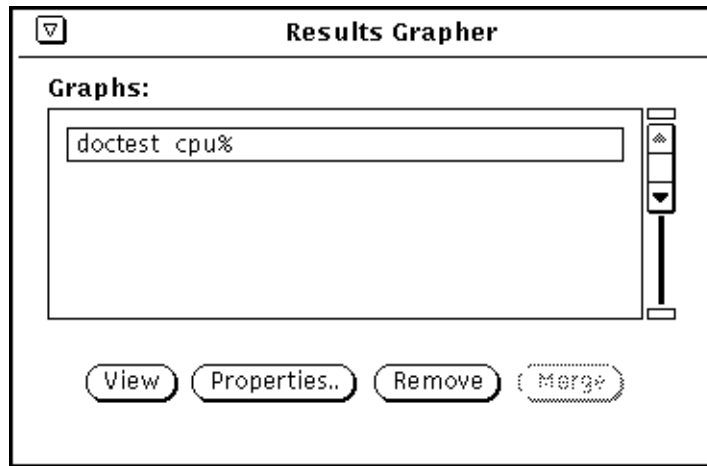


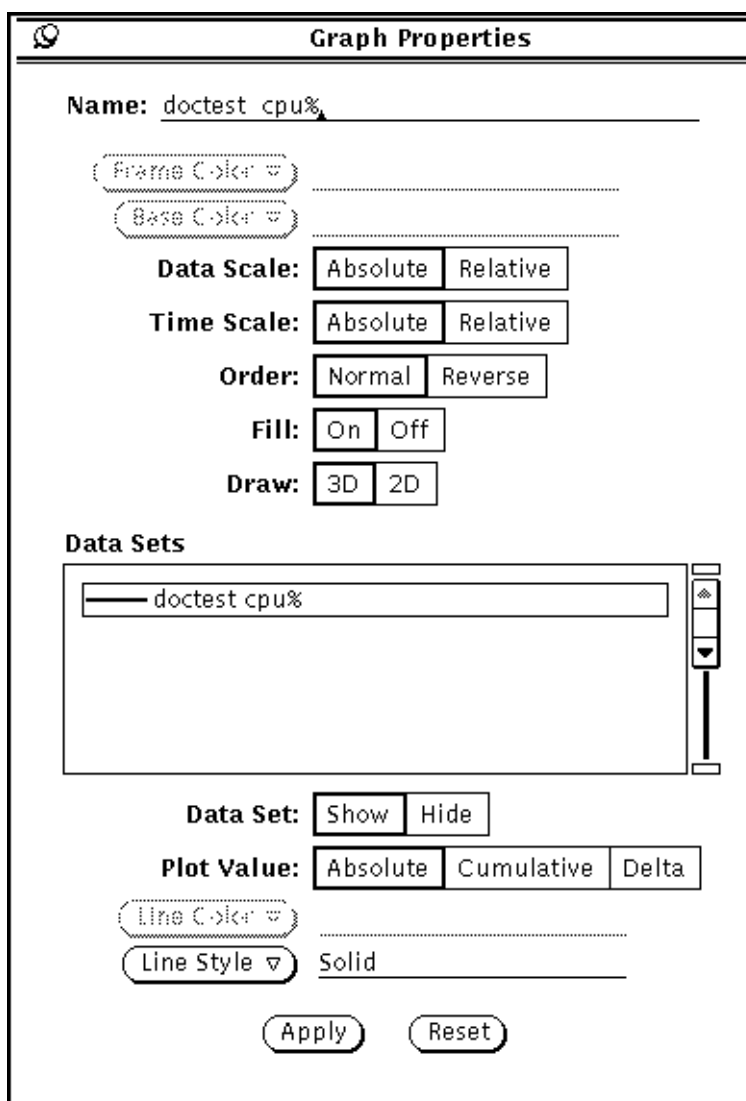
Figure 21-2 Results Grapher Window

The scrolling list contains the name of graphs that you have created. Click SELECT to select one or more of the graph names in the scrolling list. By using one of the four buttons in the window, you can perform the following operations on one or more selected graphs:

- *View* displays the selected graph in a Graph pop-up window.
- *Properties* displays the selected graph properties in a Graph Properties pop-up window.
- *Remove* deletes the selected graph, along with its properties.
- *Merge* merges the selected graphs into a single pop-up Graph window. This button is only active if there are two or more selected graphs.

21.2 Graph Properties Window

The Graph Properties window, as shown in Figure 21-3, contains parameters for the display of each graph.



Graph Properties

Name: doctest cpu%

Frame Color ▾

Base Color ▾

Data Scale: Absolute Relative

Time Scale: Absolute Relative

Order: Normal Reverse

Fill: On Off

Draw: 3D 2D

Data Sets

doctest cpu%

Data Set: Show Hide

Plot Value: Absolute Cumulative Delta

Line Color ▾

Line Style ▾ Solid

Apply Reset

Figure 21-3 Graph Properties

The following describes the Properties parameters:

Name:

Specifies the name of the graph that will appear in the scrolling list in the Results Grapher window, and as the label on the Graph pop-up window. By default, this field contains the host name, followed by the attribute selected to be graphed. If you are graphing data from multiple systems (for example, you select multiple Browser report streams to be plotted on the same graph), the name would contain multiple host names instead of a single name.

Frame Color:

Specifies the color of the frame of the graph. By default, this value is Cyan on color monitors. To change the frame color, press MENU over the Frame Color abbreviated menu button and release the pointer on the desired color in the palette. Note that on monochrome monitors, this parameter and any other fields that specify color choices are dimmed.

Base Color:

Specifies the color of the base of the graph. By default, this value is Dark Orchid on color monitors. To change the base color, press MENU over the Base Color abbreviated menu button and release the pointer on the desired color in the palette.

Data Scale:

Specifies whether the data sets (vertical axis) are plotted using Absolute or Relative values. This is useful only if more than one data set is to be plotted. If Absolute is chosen, the maximum and minimum values of the graph assume the greatest and least values among all the data sets. If Relative is chosen, the maximum and minimum values of the graph vary according to the maximum and minimum values of each data set. Absolute data scale should be used when displaying multiple data sets whose data values are of the same units—for example, CPU percentage for multiple systems. Relative data scale should be used when displaying multiple data sets whose data values are of different units—for example, CPU percentage, interrupts, and collisions for a single system. By default, Absolute is selected.

Time Scale:

Specifies whether the time stamps of the data sets are plotted in Absolute or Relative scale. If Absolute is chosen, the start and end times of the graph equal the earliest and latest times, respectively, of the data sets. If Relative is chosen, the start and end times of the graph vary according to each data set. Absolute time scale is useful when displaying data sets from reports whose time spans overlap—for example, yesterday's CPU percentage for three different systems. Data from a very short time period that is graphed with

data from a different time (for example, 30-second time slices taken six months apart) can appear almost invisible when graphed in Absolute scale. Relative time scale is useful in comparing data sets whose time stamps differ greatly—for example, CPU percentages for a single system on three different days. With Relative time scale, the time stamps at the bottom of the graph display only the relative time from the first data value. By default, Absolute is selected.

Order:

Specifies the order in which the data sets are drawn if there is more than one data set in the graph. If Normal is chosen, the graph in the foreground corresponds to the first data set in the list of the Properties window. If Reverse is chosen, the graphs are drawn in reverse order, that is, the graph in the foreground corresponds to the last data set in the list of the Properties window. By default, Normal is selected.

Fill:

Specifies the style in which the graph is to be drawn. If On is chosen, each graph is drawn in a solid fashion with the base of the graph extending up to the apex. If Off is chosen, each graph is drawn in a ribbonlike or line fashion. Setting Fill to Off allows you to view graphs that might otherwise be hidden by the graph in the foreground. By default, On is selected.

Draw:

Specifies whether the graph is drawn in a two-dimensional (2-D) or three-dimensional (3-D) manner. By default, 3-D is selected.

Data Sets

A scrolling list containing all the data sets plotted in the graph. Each item in the list has a color glyph and the host name and attribute graphed. The color glyph corresponds to the color (or line-style for monochrome monitors) used to draw the graph and may be used as a legend for the graph. This list is an exclusive list; only one item may be selected at any time.

The following four properties affect only the selected item in the Data Sets scrolling list:

Data Set:

Controls the visibility of the selected data set's graph. Show makes the data set's graph visible, while Hide makes the data set's graph invisible. By default, Show is selected.

Plot Value:

Specifies the data sets to be plotted in one of three modes: Absolute, which plots the values of the data sets as is; Cumulative, which sums up each successive data value and plots them; and Delta, which takes the difference between each pair of data value and plots them. By default, Absolute is selected.

Line Color:

Sets the line color of the graph. The Grapher automatically selects an initial color based on the number of data sets plotted.

Line Style:

Sets the line-style of the graph. This option is only active when using a monochrome monitor. One of four line-styles may be chosen: Solid, Dotted, Dashed, Dot-Dashed. By default, Solid is chosen.

To establish or change the attributes of the graph, click the Apply button. To get back the original settings of the properties just after the last Apply, click the Reset button.

Note – Changes to the default values of the Graph Properties fields *cannot* be saved from one Grapher session to another. When you exit a Results Grapher session, all changes you have made to the default values of the Graph Properties fields are lost.

21.3 Displaying Graphs

To display a graph, select it from the scrolling list in the Results Grapher window and click SELECT on the View button. A pop-up window appears as shown in Figure 21-4.

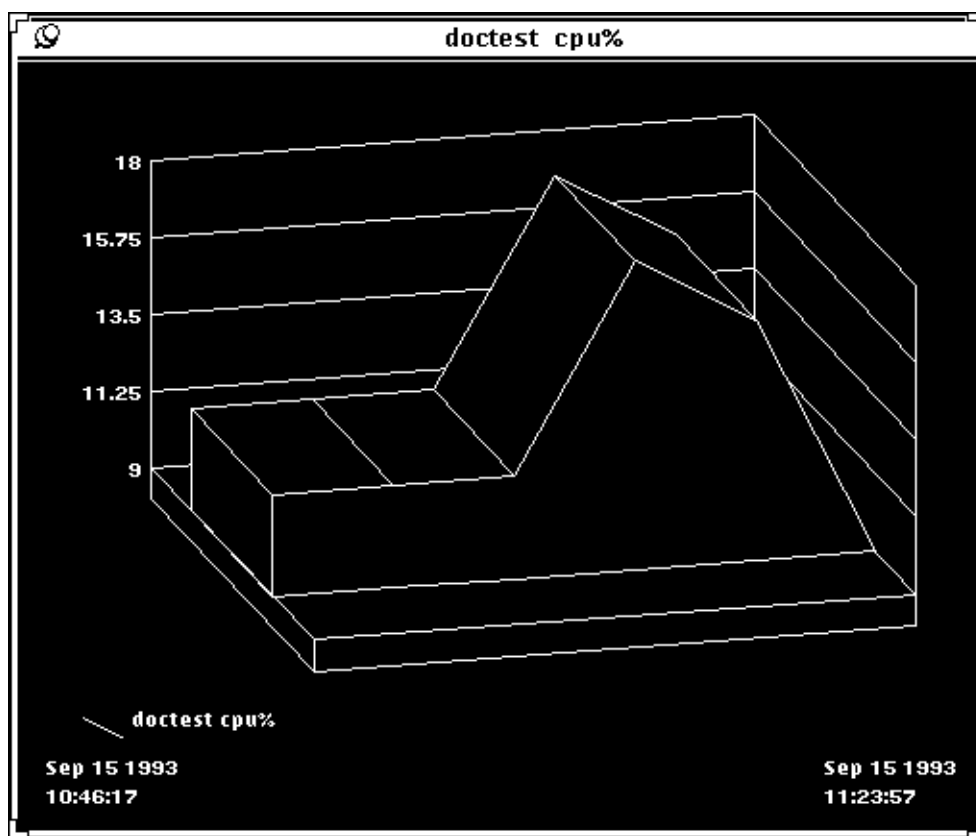


Figure 21-4 Graph

By default, graphs are plotted on a black background. To change the background to white, see the explanation on invoking the Results Grapher at the beginning of Section 21.0.1, “Starting the Results Grapher.”.

If Data Scale is set to Absolute, labels appear on the left vertical axis indicating the scales of the data sets. No labels are displayed if Data Scale is set to Relative. The start and end times of the graph are displayed at the bottom two corners of the graph; the start time is displayed at the bottom left corner and the end time is displayed at the bottom right corner.

If Time Scale is set to Absolute, the exact start and end times are displayed. If Time Scale is set to Relative, the start time is labeled 00:00:00 (<hours>:<minutes>:<seconds>) and the end time is the time elapsed since start time (<hours>:<minutes>:<seconds>).

You can zoom into a graph to get finer readings. Press SELECT at a point in the graph, then drag the mouse to another point in the graph; this must be done along the X- (time) axis. Two boundary lines mark the points that you have selected. At the same time, two time labels appear in the middle at the bottom of the graph. The time label on the left displays the start boundary time, and the label on the right displays the end boundary time. You can use this feature to inspect the times of particular points on the graph.

To replot the graph, press MENU and move the pointer to the “Replot graph with new times” option. The graph is then replotted based on the boundary times between the two points you selected. You can zoom repeatedly to get successively finer readings. To replot the original graph, press MENU and move the pointer to the “Show entire graph” option.

You can change the viewing angles of the graph in the three-dimensional view. Press MENU and move the pointer to Controls. A pop-up window containing two slider controls appears, as shown in Figure 21-5.

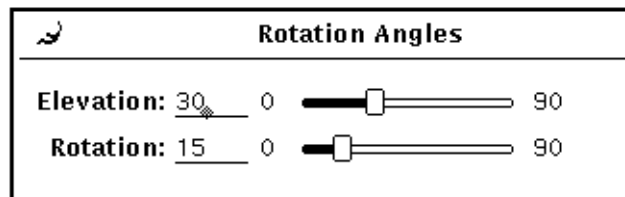


Figure 21-5 Controlling Grapher Rotation Angles

The Elevation slider controls the rotation of the graph on the vertical plane. The Rotation slider controls the rotation of the graph on the horizontal plane. You can interactively control the viewing angles of the graph by shifting the values of the sliders.

21.4 Merging Graphs

You can superimpose graphs on one another. Select the graphs to be merged from the scrolling list in the Results Grapher window, then click SELECT on the Merge button. A pop-up window appears with the selected graphs plotted together.

After graphs are merged, common parameters such as Data Scale or Time Scale take on default settings. For example, if one of the graphs being merged has Time Scale set to Relative, its Time Scale is reset to the default value of Absolute. Data set-specific parameters, such as Plot Value, retain their individual settings. Data sets are always displayed; a new line color may be assigned if there are two or more data sets using the same line color.

This chapter discusses the following topics:

- Invoking IP Discover
- IP Discover tool configuration
- `snm_discover` command
- The `discover.conf` file

The IP Discover Tool has two functions:

- A discover function, which finds hosts, routers, networks, and Simple Network Management Protocol (SNMP) devices reachable from the Console machine. On finding a network element, the discover function stores a record for that element in the runtime database.
- A monitor function, in which the tool compares the elements stored in the runtime database with the elements it finds at a specified interval or specified time. If new elements are detected, the monitor function stores the elements in a holding-area view and records these elements in a log file. Through the same log file, the monitor function can also notify you if a previously discovered host is down or was down within a given period.

When you invoke the HeadStart option at startup, IP Discover searches for a maximum of ten elements on the local subnetwork.

The hosts found by the monitor function are hosts added to a network since the last running of the discover function and hosts that the discover function did not find.

IP Discover offers a number of configuration options for both the discover and monitor functions. These options allow you to fine-tune the extent and depth of the tool's search and monitoring activities. The configuration options are described later in this chapter.

Note – When IP Discover locates SNMP devices, it cannot determine whether these devices support functionality specific to SNMPv2.

22.1 Invoking IP Discover

You invoke IP Discover using one of the following methods:

- from the Console window, when you press MENU on Tools►Discover►IP Discover and release MENU;
- from the SunOS command line, if you have the path to the SNM executables in your PATH variable, invoke `snm_discover`; for example, to run IP Discover and invoke the user interface, enter:

```
hostname# snm_discover -T
```

The path to `snm_discover` is:

- `/usr/snm/bin` on SunOS 4.x
- `/opt/SUNWconn/snm/bin` on Solaris 2.x

After IP Discover is invoked from the Console window or from the command line using the `-T` option, the window shown in Figure 22-1 is displayed. (You can start IP Discover directly, bypassing this window, if you use the `snm_discover` command without the `-T` option.)

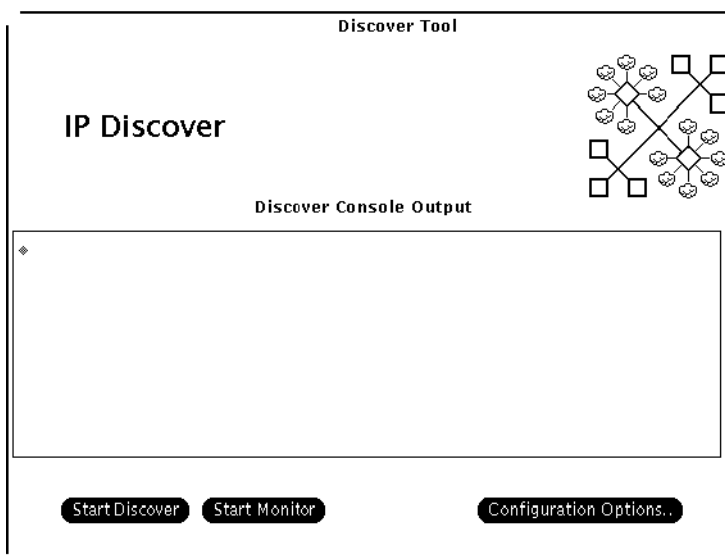


Figure 22-1 IP Discover Tool Base Window

The buttons in the IP Discover base window are described as follows:

Start/Stop Discover

This button is a toggle. When IP Discover is first invoked, Start Discover is displayed. If Start Discover is selected, the Discover function is invoked in a mode that is determined by the parameters for IP Discover in the IP Discover Configuration window (see Section 22.2, “Discover Tool Configuration”). This mode of the discover function can be limited or as extended as your time and machine resources allow. Once Start Discover is selected, the button display changes to read “Stop Discover”. Selecting Stop Discover will cause discovery operations to cease.

Start/Stop Monitor

A toggle for which the initial display is Start Monitor. This button invokes the monitor function, using the parameter values as they are set in the Monitor Properties window, described Section 22.2.3, “Monitor Function Configuration.”

Configuration Options

Invokes the IP Discover Configuration window, in which you can switch between discover and monitor function properties. These properties are discussed in the following subsections.

Once IP Discover is started, IP Discover searches for network devices in a two-stage process:

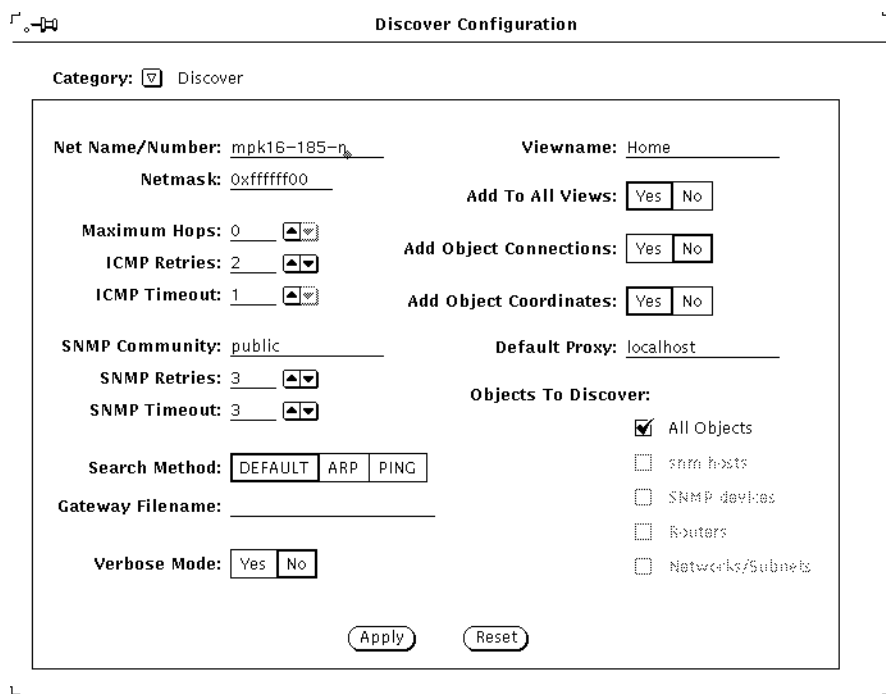
- IP Discover first builds a hierarchical linked-list structure that mirrors the network topology that it uncovers. For each Class A, B, or C network that is discovered, an entry is added to the list. Each of these entries contains a list of subnetworks (or a list of hosts directly connected to this network if the network is non-subnetted) and a list of connected networks.
- Once the network structure has been uncovered, IP Discover then issues Internet Control Message Protocol (ICMP) echo requests to find hosts on each of the subnetworks in the linked-list structure. By default, IP Discover pings each host address in the range 0 to 2048 on each subnetwork. (The command-line version of IP Discover has a `-r` option that can be used to pass IP Discover a different range of host addresses to ping. This is discussed in Section 22.4.2, “Discovery on Non-Subnetted Class B Networks.”)

To build the hierarchical network topology structure, IP Discover proceeds through the following steps:

- The local routing table is accessed to find the location of the subnetwork’s default router.
- The router’s routing table is then retrieved using SNMP.
- Routing tables are then retrieved from all next hop gateways.
- IP Discover then leapfrogs from gateway to gateway as it retrieves the routing table from each “next hop” gateway.
- IP Discover performs a traceroutes operation to determine the number of hops to each “next hop” gateway. Each router that a packet must traverse to reach that gateway counts as an additional “hop.” The traceroutes operations may also reveal gateways that were not uncovered by previous queries.
- IP Discover continues to retrieve routing tables of “next hop” gateways until the “Maximum Hops” count set in the IP Discover Properties sheet is reached. By default, “Maximum Hops” is set to zero and the discovery process is limited to the local subnetwork.
- SNMP is used to obtain the routing tables of any additional gateways that were uncovered by tracing the route to gateways previously uncovered.

22.2 Discover Tool Configuration

When you click SELECT on the Configuration Options button in the Discover Tool base window, you receive the Discover Configuration window as shown in Figure 22-2.



Discover Configuration

Category: Discover

Net Name/Number: mpk16-185-n
Viewname: Home

Netmask: 0xffffffff
Add To All Views: Yes No

Maximum Hops: 0
Add Object Connections: Yes No

ICMP Retries: 2
Add Object Coordinates: Yes No

ICMP Timeout: 1
Default Proxy: localhost

SNMP Community: public
Objects To Discover:
 All Objects
 snmp hosts
 SNMP devices
 Routers
 Networks/Subnets

SNMP Retries: 3
Search Method: DEFAULT ARP PING

SNMP Timeout: 3
Gateway Filename:

Verbose Mode: Yes No

Apply Reset

Figure 22-2 Discover Configuration Window:

To view the two categories of properties, Discover and Monitor:

1. Press MENU over the Category button.
2. Release MENU over the category of your choice.

The parameters for these categories are discussed in the following subsections. When either Discover or Monitor is active, you must stop the function before changing any properties.

22.2.1 IP Discover Function Configuration

The parameters in the IP Discover Properties window are described as follows:

Net Name/Number

The name or subnetwork number—as specified in an NIS map, an NIS+ table, or the `/etc/networks` file—that identifies the subnetwork that will be the starting point for the discover function. By default, the discover uses the network number used by the local machine. When specifying a network number, be sure to enter a value for each byte within the number. For example, specify `129.144.41.0`, not `129.144.41`.

If you enter a network name or number outside of the local subnetwork, the discover function limits its search to the subnetwork you specify.

Netmask

The network mask—as specified in an NIS map, an NIS+ table, or the `/etc/netmasks` file—used in the network(s) on which the discover function is to operate. By default, the discover function uses the netmask number used on the local subnetwork. Enter the netmask number in hexadecimal (preceded by `0x`) or dotted decimal notation (for example, `255.255.255.0`).

To limit the number of elements found, IP Discover uses the inverse of the host portion of the subnet mask. For example, a netmask of `0xfffff00` has a host portion of `0x00`, which becomes `0xff` or decimal 255.

The netmask parameter has two uses: one is for when you want discovery to begin in a subnetwork other than the local subnetwork. The discover function needs to know the netmask used on that subnetwork. The second use is as a limit to the number of hosts the discover function will add to the runtime database. The discover function will ping a maximum of 2048 network elements per subnetwork.

Maximum Hops

A measure of how far the discover function will extend its search. For a given route from the machine running the IP Discover Tool (or from the subnetwork specified in the Net Name/Number parameter), this parameter is the maximum number of routers the ICMP packets sent by the discover function will traverse. The default for this parameter is zero hops, which

means searches are restricted to the local subnetwork. By setting Maximum Hops to 1 or greater, IP Discover will extend its search to additional subnets connected to the local subnetwork's router.

Note – If there are multiple routers attached to the local subnetwork, the number of subnets searched may increase exponentially if Maximum Hops is set higher than zero. Depending on the size of your network, setting Maximum Hops to any value above zero can result in very long searches.

ICMP Retries

The discover function uses the Internet Control Message Protocol (ICMP) to find network elements that are not SNMP devices. This parameter determines the number of times the tool will retry “finding” an element using ICMP following an initial non-response. If your network is very busy, you might want to increase this value.

ICMP Timeout

Determines how long (in seconds) the discover function waits after sending an ICMP packet that did not receive a response, before sending another ICMP packet. To speed up the discovery process, reduce this value.

SNMP Community

The discover function limits its search for SNMP devices to devices that have the community name(s) you specify here. You can specify up to five SNMP community names. Use a colon to delimit multiple names. Note that using multiple community names makes the discover function take a longer time than it would with a single name. This is because the discover function tries each device with each community name until a response is received.

To make the search faster, you can use the following process:

If you are aware of the community strings for specific gateways or hosts, you can enter the information in the `snmp.hosts` file in the database directory using the format provided in the header of the `snmp.hosts` file. When you specify the community string for known gateways or hosts in this file, IP Discover only uses the read community string entered for that host. The result is a faster search because the correct string is used rather than requiring the search process to search each string entered in the SNMP community field (or in the `-c` option in the command line). If the community strings you entered are valid for only specific gateways or hosts, it is very important to use this process in order to avoid a search of each device, using each community name, until a response is received.

SNMP Retries

For SNMP devices, the discover function uses SNMP for discovery purposes. This parameter determines the number of times the tool will retry contacting an element using SNMP following an initial response. This parameter comes into play primarily when querying routers. ARP and routing tables are often large and sometimes require multiple retries to obtain complete tables.

SNMP Timeout

Determines how long (in seconds) the discover function waits for a response to an SNMP packet before giving up or, for routers, sending another SNMP packet.

Search Method

Starting with version 2.3, there are three choices for search methods: Default, ARP and Ping.

Default: the default method is a combination of Address Resolution Protocol (ARP) and Ping. IP Discover first uses the ARP tables of the gateways to update the database topology, therefore allowing you to see nodes on respective segment views. Next, ICMP echo packets are sent to locate potential devices for the missing IP addresses not found in the gateway ARP tables.

ARP: ARP search method uses ARP tables of the gateways and hosts in the network/subnet to update the database topology. An ARP table is a mapping of IP addresses to physical addresses. This search method finds hosts/routers that have recently generated traffic. Frequent discovers are, therefore, recommended.

Ping: The Ping search method sends ICMP echo requests, serially, to all possible addresses. Of the three search methods, this is the most time consuming. Command line options you can use starting with version 2.3 include the following:

-*N* <num>: the maximum number of outstanding simultaneous pings per interval specified by the -*N* option. The default is 1.

-*F*<num>: Fast Ping retries - the number of times IP Discover tries to contact a device. The default value is 1. -*F* is valid only when -*N* is greater than 1.

-*f*<seconds>: Fast Ping timeout - the frequency between transmissions of the batch of ICMP echo requests (in seconds). The default is 3 seconds. -*f* is valid only when -*N* is greater than 1.

If you select the -*N* option, you can set the ICMP response timeout and retry values using the -*F* and -*f* options.

Gateway File

You can specify the gateways to be used for the query phase. The command line format is -*G*<gateway_filename>. This is a text file containing the name of the list of gateways, one gateway name per line. In addition, you can specify the ONLY flag as #ONLY with the '#' sign in the first column in the beginning of the gateway file. There is no default gateway file. If a gateway file is not specified, this option is not enabled.

Specifying the ONLY flag will discover the default local subnet and all the networks known to the gateways listed in the gateway file, irrespective of the user-specified hop count.

If you do not specify the ONLY flag, the networks discovered are the local subnet, all the networks known to the listed gateways (even if the hop count does not match), and all the networks within the user specified hop count. Following is sample gateway file

```
#ONLY
mpk16router
mpk17router
```

Verbose Mode

A yes/no switch for which the default is no. Click SELECT on Yes to have the discover function report detailed progress in the IP Discover Console Output window in the IP Discover base window. A no selection means that the discover function reports only intermittent progress.

We recommend a yes setting for this parameter, particularly for long searches.

Viewname

Name of the view to which the discover function will add discovered network elements. If Add To All Views (see below) is set to no, the discover function preserves a separate view hierarchy under this name. If the name you specify here is not in the runtime database, the discover function creates a new view and adds discovered elements to it. The name specified here will be appended to all subviews in this view hierarchy.

The SNM database does not permit duplicate names of views (or other elements). By default, IP Discover uses network and subnetwork names in building the database representation of the network topology. If you want to have different views into the same subnetwork or network — such as a view of subnetwork A that shows all hosts and a view of subnetwork A that shows only routers — then the names specified for these views must be distinct. By appending Viewname to the names of the subviews in a view hierarchy, the separateness of the subviews in that view hierarchy is preserved. IP Discover will append Viewname to the names of subviews only if Add To All Views is set to “No.”

Add To All Views

A yes/no switch for which the default is yes. If you accept the default of Yes, the discover function adds discovered elements to all views in the runtime database, including the view specified in Viewname, if one is present. If you click SELECT on No, the discover function appends the name specified in the Viewname field to all views (that is, elements of category view) within the specified view. For an element that is in multiple, high-level views, this feature allows you to distinguish that element as it exists in different views, such as a router that is in a routers-only view of a subnetwork but also in a view that shows all devices on that subnetwork. Since the SNM database does not allow views (or other elements) with duplicate names, you can only have multiple views into the same

subnetwork if you assign separate names to maintain the uniqueness of each view. The use of this parameter to create separate view hierarchies is illustrated in Section 22.2.2, “Example: Creating a Routers-only View.”

Add Object Connections

A yes/no switch for which the default is no. Click SELECT on Yes to have the discover function create manageable connections between discovered elements and to add the connection information to the `linkmap` file. However, the discover function only adds connection information to the `linkmap` file if Maximum Hops has been set to 2 or more. Refer to the `linkmap (5)` manual page for information about the format of the `linkmap` file and its usage. A No selection means that the discover function does not make connections.

Add Object Coordinates

A yes/no switch for which the default is yes. By default, the discover function performs minimal layout of elements within a view. For example, in a subview representing an Ethernet subnetwork, glyphs representing hosts are distributed on both sides of the bus glyph. If you select no,

elements are created with no layout of elements taking place. A sample subview created without coordinates is shown in Figure 22-3.

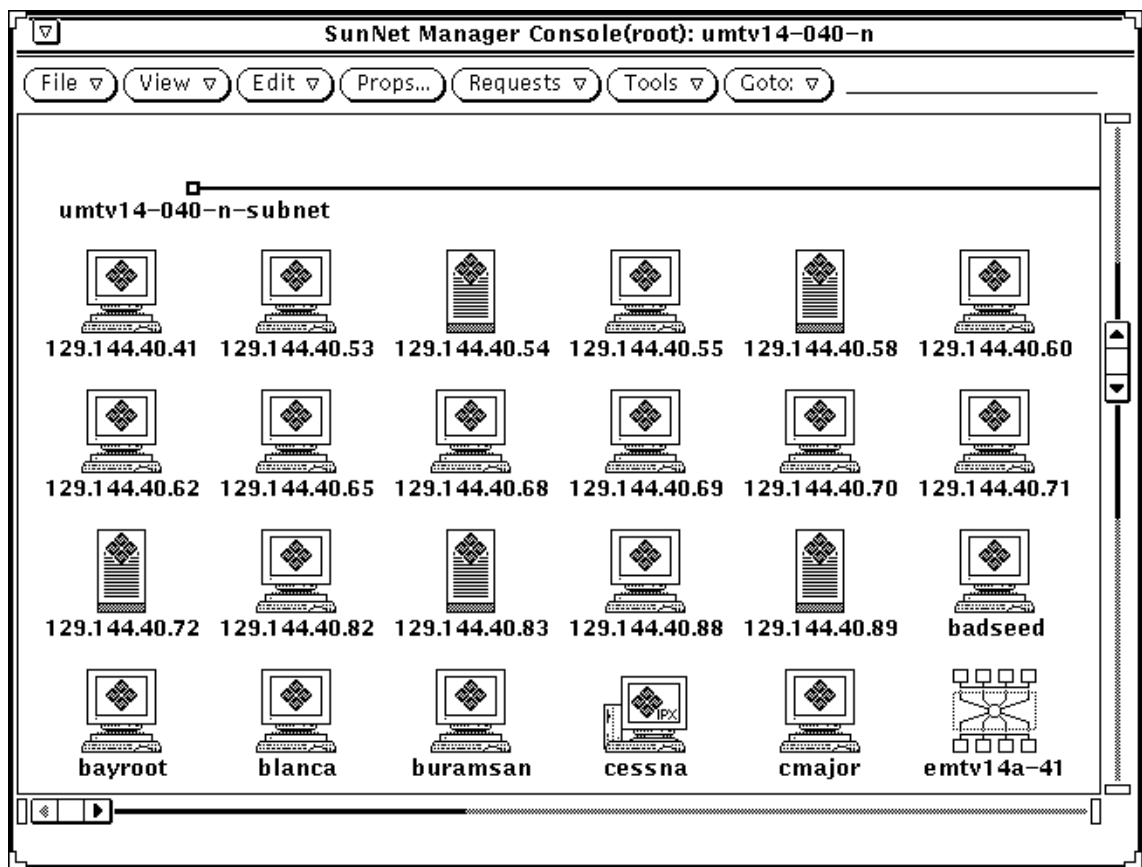


Figure 22-3 Subview Created without Coordinates

Default Proxy

When the discover function finds an element that has SNMP proxy agents, it specifies the host you enter here as the proxy host. If you leave this item blank, the proxy host is specified as localhost.

Objects to Discover

By default, the All Objects item is selected. If you click SELECT on the All Objects checkbox to deselect it, the following dimmed items display normally:

snm hosts

Systems running SunNet Manager agents.

SNMP devices

Systems or other devices running SNMP. Note that the discover function cannot discover whether the SNMP device supports the added functionality of SNMPv2.

Routers

Systems that have multiple network interfaces to different IP networks. When the discover function finds a router, it fills in IP addresses for the first two interfaces and uses the alias feature to give a name to each interface (if, indeed, each interface has a unique name), including those beyond the first two. You can then specify any of these aliases when sending a data or event request.

Networks/Subnets

Collections of subnetworks or individual subnetworks. If you select this item, the discover function finds only networks and subnetworks, plus manageable connections and routers if the Add Object Connections item is selected.

Accept the default of All Objects or click SELECT on the checkbox(es) for object(s) of your choice.

22.2.2 Example: Creating a Routers-only View

You might want to use IP Discover to create a separate view hierarchy — named “Routers” — that contains only routers in addition to another view hierarchy that contains all objects. To create a routers-only view hierarchy, you would need to make the following entries (as shown in Figure 22-4) in the IP Discover Configuration window:

- Specify “Routers” in the Viewname field
- Set Add To All Views to No
- Select Routers only in the Objects To IP Discover field.

Discover Configuration

Category:

Net Name/Number: Viewname:

Netmask: Add To All Views: Yes No

Maximum Hops: Add Object Connections: Yes No

ICMP Retries: Add Object Coordinates: Yes No

ICMP Timeout: Default Proxy:

SNMP Community: Objects To Discover:

SNMP Retries: All Objects

SNMP Timeout: snm hosts

Search Method: DEFAULT ARP PING SNMP devices

Gateway Filename: Routers

Verbose Mode: Yes No Networks/Subnets

Figure 22-4 Example of Routers-Only IP Discover Configuration

A glyph named “Routers” will be added to the Home view. If there are multiple subviews under “Routers” that represent subnetworks, the names of these views will have “-Routers” appended to them, as shown in Figure 22-5.

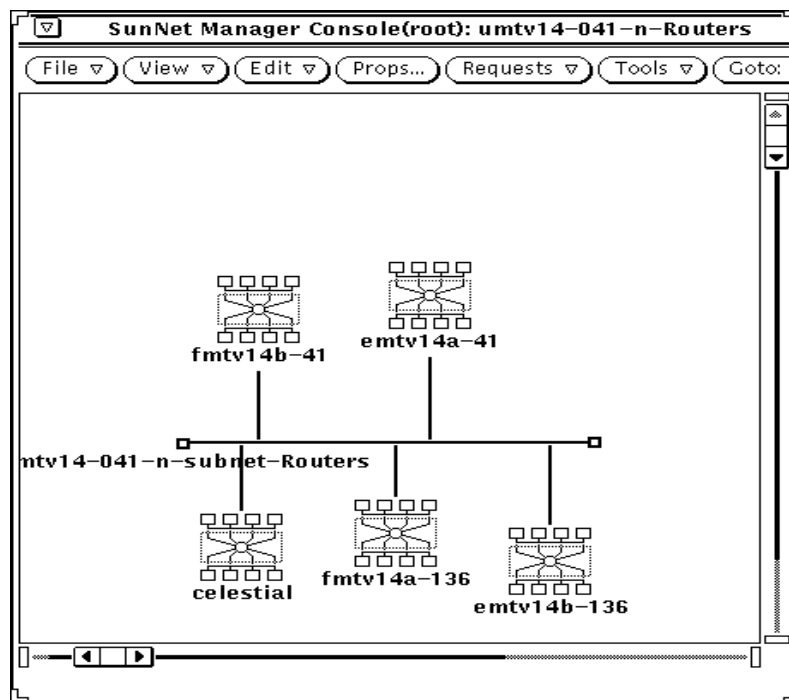


Figure 22-5 Subview in Routers-only Hierarchy

22.2.3 Monitor Function Configuration

The window for monitor function configuration is the window you receive when you press MENU in the Category button in the IP Discover Configuration window and release MENU over Monitor. The window shown in Figure 22-6 is displayed.

Monitor Configuration

Category:

Holding Area Viewname: Verbose Mode: Yes No

Non-response Timeout: / min Start Time:

Time Between Cycles: / min Stop Time:

ICMP Timeout: / sec

Log History: Yes No Start Date:

Log Filename: Stop Date:

Mail Logfile To:

Ignore-host Filename: Run Weekly: Yes No

Day of Week:

Figure 22-6 IP Discover Configuration: Monitor Window

The parameters in the window above determine the operation of the monitor function only; they have no effect on the discover function. (However, the ICMP and SNMP-related discover parameters *do* apply to the monitor function.)

By default, the monitor function works silently. Set the Verbose Mode option (described below) to Yes to have the function display messages in the IP Discover Tool Console Output window. Also by default, the monitor function runs daily; you can set it to run weekly instead, by setting the Run Weekly and Day of Week parameters, described below.

To protect the integrity of already-discovered views, the monitor function adds hosts to a “holding area” view you define in the property sheet (see Holding Area Viewname, below).

The monitor parameters in the IP Discover Properties window are described as follows:

Holding Area Viewname

The name of the view to which the monitor function adds the hosts it finds.

Non-response Timeout

Determines how long (in minutes) the monitor function will continue to try to contact an unresponsive host before marking the host as down in the file specified in the Log Filename item (described below).

Time Between Cycles

Within the span specified by Start Time and Stop Time (see below), this parameter determines how long (in minutes) the monitor function will wait between monitor cycles. A monitor cycle is one complete pass of the monitor function through the network elements in a runtime database. A value of 0 means that the monitor function runs continuously until the Stop Time is reached.

ICMP Timeout

Determines how long (in seconds) the monitor function waits after sending an ICMP packet before sending another ICMP packet.

Log History

An on/off switch the default for which is on. Determines whether the monitor function keeps a record of its operation. You must set this parameter to On to have data sent to a log file and to have to the log file sent to an electronic mail address.

Log Filename

Full path name of the file in which the monitor function stores a record of its operation. Note that you must specify a valid path name for this parameter to obtain a record of monitor output.

Mail Logfile To:

An electronic mail address to which the monitor function sends a record of its operation upon completion of a cycle. The software performs no error checking on the address you enter.

Ignore-Host File Name

Full path name of the file listing names of hosts and views that you want the monitor function to ignore. The requirements for the file are that it be an ASCII file with one hostname or view per line, no blank lines, with host names beginning in column 1. The monitor function does not look inside a view listed in this file.

You might find the ignore-views feature useful if you have object-type-specific views. For example, if you have a router-only view, the monitor function, left unchecked, adds all of the hosts it finds on the routers' subnetworks to this view. If you insert the name of this view in the file you specify for the Ignore-Host File Name, the monitor function will not look inside this view.

Use the ignore-hosts feature if the monitor function frequently finds new or down hosts that are of no interest to you.

Verbose Mode

A yes/no switch for which the default is no. Click SELECT on Yes to have the monitor function report progress in the IP Discover Console Output window in the IP Discover base window. A no selection means that the monitor function works silently. A no selection for this parameter does not affect the logging-related parameters.

Start Time

A time at or after which the monitor function begins operation within the specified start and stop dates. The initial display is the current time. The start time takes effect for daily or for weekly use of the monitor function.

Stop Time

A time at which the monitor function ceases operation within the specified start and stop dates. The stop time takes effect for daily or for weekly use of the monitor function.

Start Date

A date on or after which the monitor function begins operation. The initial display is the current date. The start date takes effect for daily or for weekly use of the monitor function.

Stop Date

A date on which the monitor function ceases operation. The stop date takes effect for daily or for weekly use of the monitor function.

Run Weekly

A yes/no switch for which the default is no. A no selection means that the monitor function works on a daily rather than a weekly basis. When this item is set to no, the following item (Day of Week) is greyed-out. Click SELECT on Yes to have the monitor function run on a weekly rather than a daily basis.

Day of Week

If the previous item is set to yes, this item displays normally. The initial selection is the current day of the week. Press MENU in the abbreviated menu button to obtain a day-of-the-week menu and release MENU over the day on which you want to monitor function to run.

22.3 Updating the Management Database

Use IP Discover's *Monitor* function to schedule periodic searches for new additions to the network or nodes not uncovered previously. Use this tool to also build a hierarchy of views representing your network topology.

Rather than burden the network with frequent IP Discover searches, Monitor might be scheduled to run once a week. To access the Monitor Configuration window do the following:

- 1. Press MENU on the Tools button on the Console control panel to pull down the Tools menu.**
- 2. Select Discover.**
- 3. Select IP Discover.**
- 4. When the IP Discover window appears, click SELECT on the Configuration Options button.**
- 5. Press MENU on the Configuration Category button and select Monitor.**

When Monitor discovers new devices, it adds them to a holding area view. In Figure 22-7, this view is named "New_Devices." In this example Monitor is configured to run weekly. You can select the day of the week and the time period in which Monitor will run.

Refer to information on configuring Monitor in the "IP Discover" chapter in "Part 2: Reference."

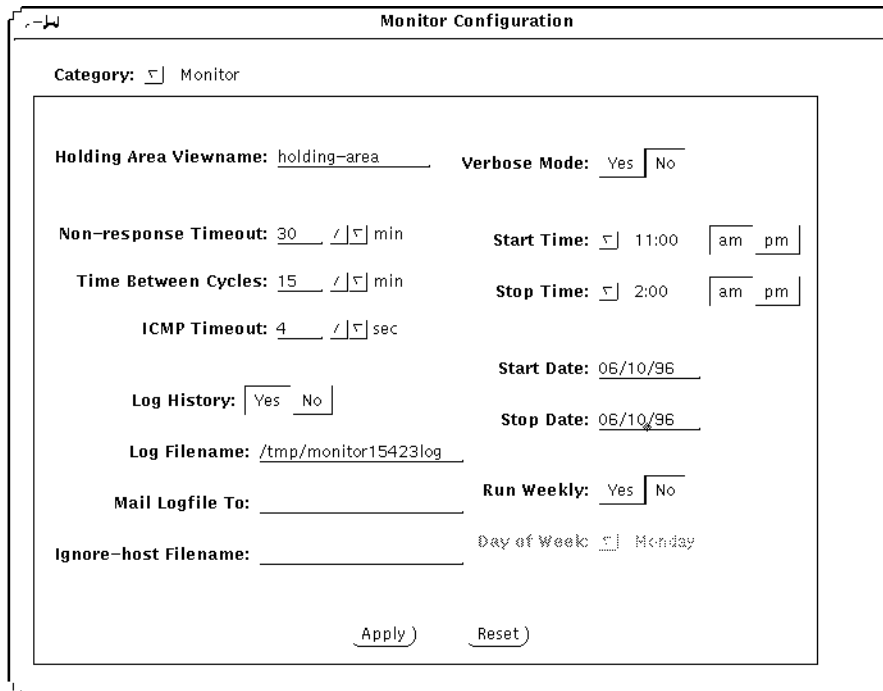


Figure 22-7 Example of Monitor Configuration

22.4 snm_discover Command

You can invoke the IP Discover Tool from the SunOS command line using the `snm_discover` command. Many of the options that you can use with this command correspond to fields in the IP Discover or Monitor menus of the IP Discover Configuration popup window. If you have installed the Solaris 1.1 version of the current product, you enter:

```
hostname% /usr/snm/bin/snm_discover <options>
```

If you have installed the current product on a Solaris 2.x machine, you enter:

```
hostname% /opt/SUNWconn/snm/bin/snm_discover <options>
```


22.4.1 IP Discover Command-Line Options

- A
specifies that the search method is ARP only. Refer to Section 22.2.1, “IP Discover Function Configuration,” on page 22-6 for information on the ARP search method.
- c <community>
<community> specifies the SNMP community string to use. Equivalent to the “SNMP Community” field in the Configuration window. Refer to Section 22.2.1, “IP Discover Function Configuration,” on page 22-6 for information on the SNMP community.
- d <objects>
By default, IP Discover searches for all objects. The -d option limits the discovery to types of objects specified in a list. Possible entries: routers, snmp, snm, or network. Entries in the list are separated by colons. Equivalent to the “Objects to Discover” field in the Configuration window.
- F <num>
Fast Ping retries - the number of times IP Discover tries to contact a device. The default value is 1. -F is valid only when -N is greater than 1 and the -P option is selected.
- f <seconds>
Fast Ping timeout - the frequency between transmissions of the batch of ICMP echo requests (in seconds). The default is 3 seconds. -f is valid only when -N is greater than 1 and the -P option is selected.
- h <hops>
<hops> measures how far IP Discover will search. By default this value is 0; in that case, IP Discover limits its search to the local subnet. For a given route to a possible target, <hops> measures the number of routers that will be traversed by packets sent by IP Discover. This is equivalent to the “Maximum Hops” field in the Configuration window.
- G <gateway filename>
Use this option to specify the gateway. Refer to Section 22.2.1, “IP Discover Function Configuration,” on page 22-6 for information on the gateway filename.
- H
Use this option to display IP Discover command line options.

- l
Manageable connections are created between discovered elements and connection information is added to the `linkmap` file. Equivalent to the “Add Object Connections” field in the Configuration window.
- m *<netmask>*
<netmask> specifies the netmask used in IP Discover. This can be in hexadecimal or dotted decimal notation. Equivalent to “Netmask” field in Configuration window.
- M
Starts Monitor.
- N *<num>*
the maximum number of outstanding simultaneous pings per interval specified by the -N option. The default is 1.
- n *<network>*
<network> specifies the name or subnetwork number—as specified in an NIS map, an NIS+ table, or the `/etc/networks` file—that identifies the subnetwork that will be the starting point for the discover function. By default, the discover uses the network number used by the local machine. Equivalent to the “Net Name/Number” field in the Configuration window.
- o
When a *<viewname>* is specified with the -v option, and the -o option is also specified, IP Discover adds elements only under the view specified in *<viewname>*. Subviews created under this view have *<viewname>* appended. This is equivalent to entering “No” for the “Add to All Views” field in the Configuration window. An example illustrating the use of this option is discussed in Section 22.2.2, “Example: Creating a Routers-only View.”
- P
Use this option to specify that the Ping search method should be used. Refer to Refer to Section 22.2.1, “IP Discover Function Configuration,” on page 22-6 for information on the Ping search method.
- p *<num>*
<num> specifies the number of times IP Discover will reissue an ICMP echo request to a host address after an initial non-response. Equivalent to “ICMP Retries” field in Configuraion window.

- q
This invokes a quick IP Discover, which limits discovery to a maximum of ten devices on the local subnet.
- r <IPaddress1>:<IPaddress2>
By default, IP Discover only pings host addresses in the range 0 to 2048 on each subnet that it finds. The -r option passes a range of IP addresses to use as targets for ICMP echo requests in searching for hosts. The use of this option is discussed in the next section.
- s <seconds>
<seconds> specifies the number of seconds to wait for a response to an SNMP packet before sending another SNMP packet. Equivalent to the “SNMP Timeout” field in the Configuration window.
- S <num>
The number of SNMP retries when sending requests to routers. Equivalent to the “SNMP Retries” field in the Configuration window.
- t <seconds>
Determines how many seconds IP Discover waits before sending another ICMP request after non-response to a prior request. Equivalent to “ICMP Timeout” field in Configuration window.
- T
Invokes the IP Discover Tool window interface.
- v
Turns on verbose mode. Equivalent to setting “Verbose Mode” field to “Yes” in Configuration window.
- V <viewname>
Puts all discovered devices into <viewname>. By default, <viewname> is “Home.” Equivalent to the “Viewname” field in the Configuration window.
- x
By default, IP Discover does some minimal layout of elements in views. If this option is used, elements are created without screen coordinates and no layout of elements takes place.

22.4.2 Discovery on Non-Subnetted Class B Networks

When IP Discover attempts to find elements on a network, ICMP echo requests will be sent to a maximum of 2048 host addresses on each subnetwork. By default, these host addresses will be those in the range 0 to 2048. However, non-subnetted Class B networks may have host addresses higher than 2048.

In a Class B IP network address, two octets (16 bits) of the four-octet (32-bit) IP address are available as host addresses. In a subnetted Class B network, a portion (typically one octet) of this 16-bit host address is used for subnet addresses.

On a non-subnetted Class B network, the entire 16-bit host portion of the IP address is available for host addresses, thus allowing host addresses higher than 2048. (Host addresses higher than 2048 can also occur on subnetted Class B networks if 4 or fewer bits of the host portion of the IP address are used for subnet addresses. In general, any subnetted Class A or Class B network that has 12 or more bits available for the host address can have host numbers higher than 2048.)

Although an individual IP Discover session will only ping 2048 devices on each network, the possible range of host addresses is not limited to the default 0–2048. The IP Discover `-r` option can be used to modify the range of addresses to locate devices with host addresses higher than 2048 on non-subnetted networks. This is done by invoking IP Discover with the `-r` option at the SunOS command line:

```
hostname# <snm_path>/snm_discover -r <IPaddress1>:<IPaddress2>
```

By default, `<snm_path>` is `/usr/snm/bin` for SunOS 4.x installations or `/opt/SUNWconn/snm/bin` for Solaris 2.x installations.

Progressively higher blocks of host addresses can be passed to IP Discover in subsequent sessions to exhaustively search the range of possible host addresses on large non-subnetted networks.

22.5 *The discover.conf File*

The material in this section is for those users who:

- Want to modify the shape or the color of the glyphs that represent the elements found by the IP Discover Tool;
- Want to use system descriptions returned by SNMP agents to specify which type of element to create
- Have created customized views that you want to monitor
- want to use the IP Discover Tool to find agents in addition to the agents shipped with SNM.
- Associate agent schemas with components

The `discover.conf` file is an ASCII file stored in the following directory

- `/var/adm/snm` on Solaris 1.x machines
- `/var/opt/SUNWconn/snm` on Solaris 2.x machines

The file contains individual sets of configurable fields, each set identified by the following uppercase keywords:

- `OID`
- `COLORS`
- `COMPONENTS`
- `DEFAULTS`
- `MAPPINGS`
- `MONITOR COMPONENTS`
- `AGENTS`
- `AGENT SCHEMA MAPPING`

In the `discover.conf` file, each of these keywords must be preceded by a hash mark (`#`). Except when preceding a keyword, a hash mark indicates a comment.

Within `discover.conf`, the order of fields *is* significant.

22.5.1 *OID Section*

The OID section maps object identifiers (OIDs) to element types. An element type determines the glyph used for a given element. An entry has the form `<oid> <element_type>`, where `<element_type>` is in a schema file that has been loaded into the Console. For example:

```
#OID
#
Sun Microsystems.2.1.1 component.ss10
HP.2.3.2.5 component.hpsnake
```

When a device is reachable through SNMP, the discover function tries to retrieve the OID for that device. If successful, the discover function maps the OID to a specific SNM element type. In the example above, any device returning the OID for Sun (which can be in either the textual form of the example or the numeric 1.3.6.1.4.1.42.2.1.1.1) is created as the SNM element type `component.ss10`. Using the same example, if the device returns HP.2.3.2.5 the element type is `component.hpsnake`.

22.5.2 *COLORS Section*

The COLORS section of `discover.conf` allows you to create your own color names for the two sections, COMPONENTS and DEFAULTS, that follow COLORS. As shipped with SNM, the COLORS section is as follows:

```
#COLORS
#
RED          255 0 0
BLUE         0 0 255
ORANGE       255 0 255
YELLOW       255 255 75
```

See the bottom area of an element’s property sheet for slider bars that show values for different colors.

22.5.3 COMPONENTS Section

The COMPONENTS section of `discover.conf` associates element types with the colors defined in the COLORS section. For example:

```
#COMPONENTS
#
component.ssl          ORANGE
component.server       LIGHT_BLUE
bus.ethernet           RED
view.subnet            BROWN
view.network           DARK_GREEN
component.link         BLACK
```

22.5.4 DEFAULTS Section

The DEFAULTS section of `discover.conf` allows you to assign a specific color for all SNMP devices. For example:

```
#DEFAULTS
#
snmp_color             GREEN
```

With the entry above, glyphs for all SNMP devices will be colored green.

22.5.5 MAPPINGS Section

The MAPPINGS section allows you to map a keyword returned from the SNMP system description to an SNM element type. If a keyword has white space, it must be enclosed in quotes. The IP Discover Tool has the Sun machine types listed in Table 22-1 as defaults.

Table 22-1 Sun Machine Types

"sparcstation 10"	component.ss10
"sparcstation 1"	component.ss1
"sparcstation 2"	component.ss2
"sparcstation 330"	component.ss330
"sparcstation 370"	component.ss370
sun386	component.sun386
sun3	component.sun3
sun4	component.sun3
sun470	component.sun3
sc2000	component.sc2000
sc1000	component.sc2000
ipc	component.ipc
ipx	component.ipx
lx	component.lx

The mappings in the `discover.conf` file are consulted before the defaults shown above. For example, if the system description returned by the remote SNMP agent is:

```
SNMP agent ibm pc
```

...and you have the following entries in your `discover.conf` file:

```
#MAPPINGS
"ibm pc"          component.pc
bridge component.bridge
```

...the type for the element created by the IP Discover Tool will be `component.pc`. Note that the keyword can appear anywhere in the system description returned by the SNMP agent. Using the preceding example, if a system description is "better to build a bridge," the IP Discover Tool will create a component of the type `component.bridge`.

22.5.6 MONITOR COMPONENTS Section

The MONITOR COMPONENTS section enables the monitor function to monitor views of types other than networks and subnets. An entry in this section has the following form:

```
<view type>
```

...where *<view type>* identifies the type of view that you want to the monitor function to monitor. The view type and view instance must be successfully loaded in the Console. The view type must have a netmask field; for the view instance, the netmask field must have a non-null value.

The view types shipped with SNM (to see the list, invoke Edit►Create in the Console's control area, then click SELECT on the View category in the Create Object window) do not qualify for monitoring because they do not have a netmask field.

22.5.7 AGENTS Section

The AGENTS section enables the IP Discover Tool to search for agents in addition to those shipped with SNM. An entry has the following form:

```
<RPC number> <agent_name> [proxy]
```

The entries in this file are described as follows:

<RPC number>

The RPC number of the agent as it is specified in the `/etc/services` file or a directory service map or table.

<agent_name>

The agent name as it appears in the properties sheet for an element. Note that this is not `na.<name>`, but simply `<name>`. *<agent_name>* must identify an agent that is in a schema file that is loaded in the Console.

`proxy`

If the agent is a proxy agent, use the word `proxy` to indicate its proxy status.

22.5.8 AGENT SCHEMA MAPPING Section

The agent schema for a specific device is configurable. You can associate specific agent schemas with specific component types listed in this Section. The IP Discover application will automatically activate the agent schemas for all the device types listed in this Section along with the existing defaults. The following format must be used while using this Section:

```
#
#AGENT SCHEMA MAPPING
#
<snm element type 1>
{
    <agent schema 1>
    <agent schema 2>
    .
    .
    .
    ,agent schema n>
}
<snm element type 2>
{
<agent schema 1>
<agent schema 2>
}
```

...where *<snm element type>* are the component names such as *component.ss1*, *component.ipc* specified in the #MAPPINGS Section of this file and *<agent schema>* are the agent schemas such as *cpustat*, *diskinfo*, and so forth.

The IPX Discover feature is new for version 2.3 and offers the following functions:

- Discovers IPX nodes, clients, and servers and displays their logical topology
- Discovers IPX Netware services provided by these nodes

See the man page for IPX Discover for details about the functionality and command line options. This chapter gives an overview of how the IPX Discover process works and describes the screens.

23.1 Function Overview

IPX Discovers and models IPX networks by communicating with one or more topology export/import agent (NXIS) located on the Novell Management platform (Managewise). NXIX exports topology data discovered by Managewise; Managewise need not be running during this process. The data is then modeled appropriately, and a format topology is displayed and stored out to the database.

The NXIS script Format Specification 1.0 specifies how data is exported and protocols for communication between SunNet Manager and NXIS. NXIS listens for client connections at TCP/IP port 1486 at the NMS console.

23.1.1 Configuring the Export/Import Agent

Before you begin IPX Discover, ensure that the Export/Import Agent (NXIS) is running on the Novell Managewise console. The console need not be running, but the NetExplorer program which discovers Novell Networks must be running on the Novell Server.

To configure NXIS on the Novell server:

- 1. Click on the Export Agent Setup window.**
- 2. Enter the Console Name.**
The name you provide here must be the same name you provide in the IPX configuration setup.
- 3. Enter the Password.**
The name you provide here must be the same name you provide in the IPX configuration setup.
- 4. Console Type: Other**
- 5. Select Advance.**
- 6. Port Number: configure for 1486**
- 7. Database Extraction Schedule: Enter information specifying when information is to be extracted from the database.**
- 8. Advance window: Click OK**
- 9. Main window: Click OK**
- 10. Restart Export/Import Communication Agent.**

23.1.2 IPX MIBs and Schema

Following are the schemas related to IPX devices that support SNMP over IP.

- `nwalarm.schema`
- `nwhostx.schema`
- `nwserver.schema`
- `nwtrend.schema`
- `snmp-mibii.schema`

To begin the search:

1. Click **SELECT** on the **Console Menu Tools** button, and drag to **Discover**, then to **IP Discover**. You receive the screen in Figure 23-1.
2. Click **SELECT** on the **Configuration** button on the **IPX Discover Home** screen to **configure the search for NetWare devices**. You receive the screen shown in Figure 23-2. Enter the appropriate search information.

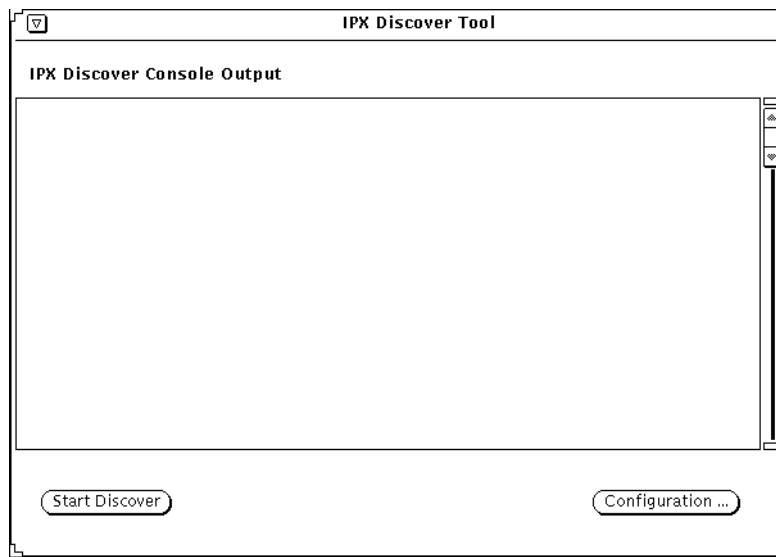


Figure 23-1 IPX Discover Home Screen

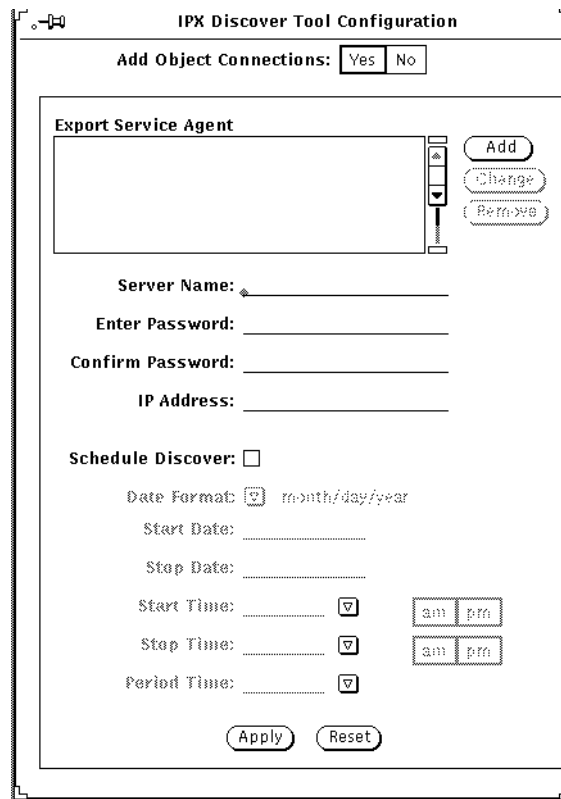


Figure 23-2 IPX Discover Configuration Screen

Note – You must choose “Other” as the option in which to send data, not IBM File Format.

23.2 Forwarding Novell’s NMS Alarms to SunNet Manager

This section describes how to configure the current release of SunNet Manager and Novell’s NetWare Management System (NMS) to forward NMS SNMP alarms from NMS to SunNet Manager over TCP/IP using the NMS Export Import Service (NXIS).

23.2.1 Software Requirements

Prior to making the configuration changes described below, verify that the following software requirements are met:

- The current release of SunNet Manager is installed.
- For the NetWare Management System server:
 - NetWare Management Agent 1.5 software is installed.
 - TCPIP NLM is loaded.
 - NW2SNMP NLM is loaded with the GA switch set to ON (“load nw2snmp ga=on”).

23.2.2 Configuration Changes Needed to SNM and NMS

To forward alarms from NMS to SunNet Manager, configuration changes are needed on both sides. NMS must be configured to send the alarms to SunNet Manager and SunNet Manager must be configured to understand and interpret the alarm information. These changes are described below.

23.2.2.1 SunNet Manager Configuration Changes

The alarms forwarded from NMS to SunNet Manager are generically referred to as “enterprise-specific traps.” In order for SunNet Manager to handle and interpret the Novell enterprise traps, Novell’s NetWare Server trap MIB (`nwalarm.mib`) must be integrated into SunNet Manager as described in the following steps. For more details, refer to “Part 2: Reference.” In the examples below `<SNM_path>` refers to the location where SunNet Manager has been installed. The default location is as follows:

- `/opt/SUNWconn/snm` for Solaris 2.x installations
- `/usr/snm` for Solaris 1.x installations

Obtain a copy of the `nwalarm.mib` file from the NetWare Management Agent Setup diskette.

4. As root, load this file into the `<SNM_path>/agents` directory.

5. As root, run the SunNet Manager `mib2schema` utility on the `nwalarm.mib` file. This will produce three files (`nwalarm.mib.schema`, `nwalarm.mib.oid`, and `nwalarm.mib.traps`) as shown below:

```
hostname# <SNM_path>/bin/mib2schema nwalarm.mib
Translating...
Translation Complete.
Schema file in "nwalarm.mib.schema"
Oid file in "nwalarm.mib.oid"
Traps in "nwalarm.mib.traps"
```

6. As root, run the SunNet Manager `build_oid` utility. This will update the object identifier database with the information from the `nwalarm.mib.oid` file as follows:

```
hostname# <SNM_path>/bin/build_oid
Parsing <SNM_path>/agents/enterprises.oid
Parsing <SNM_path>/agents/snmp-mibII.oid
Parsing <SNM_path>/agents/snmp.oid
Parsing <SNM_path>/agents/sun-snmp.oid
Parsing <SNM_path>/agents/nwalarm.mib.oid
Writing <database_path>/oid.dbase
```

7. If the `nwalarm.mib.traps` file is the only trap file that you need for your devices, you can copy the contents of that file to the default trap file. On the system where the trap daemon is running, append the contents of `nwalarm.mib.traps` to the end of the file specified in `snm.conf` as the `na.snmp-trap.default-trapfile` file. (The `snm.conf` file is located in `/etc/opt/SUNWconn/snm` for Solaris 2.x; in `/etc` for Solaris 1.x.) Refer to “Part 2: Reference” for detailed information on the trap daemon and creating an SNMP host file.

For Solaris 2.x:

```
hostname# cd /opt/SUNWconn/snm/agents
hostname# cat nwalarm.mib.traps >> /var/opt/snm/snmp.traps
```

For Solaris 1.x:

```
hostname# cd /usr/snm/agents
hostname# cat nwalarm.mib.traps >> /var/adm/snm/snmp.traps
```

or:

Create (or update) an SNMP host file by inserting the appropriate entry for the NetWare Management server host and specifying the `nwalarm.mib.schema` and `nwalarm.mib.traps` files (in the `agents` directory) as the schema and trap files respectively. Refer to “Part 2: Reference” for detailed information on the trap daemon and creating an SNMP host file.

An example entry added to the `snmp.hosts` file for the NMS server “steam” follows:

```
steam public public 10 <SNM_path>/agents/nwalarm.mib.schema
<SNM_path>/agents/nwalarm.mib.traps
```

23.2.2.2 NMS Configuration Changes

In the Netware Management system, the file `TRAPTARG.CFG` defines the recipients of SNMP traps (alarms).

This file is located in the `SYS:\ETC` directory of servers in which the NetWare Management Agent 1.5 software is installed. The file allows the user to define recipients to receive SNMP traps over IPX and UDP.

To forward traps to SunNet Manager, edit the TRAPTARG.CFG file and under the UDP section, add the IP address of the SunNet Manager machine which is to be the recipient of the SNMP alarms. In the following example, the IP address 123.123.34.56 has been added to the UDP section of the TRAPTARG.CFG file:

```
Protocol UDP
# In this section you can put SNMP managers that want to receive
# traps from the local node over UDP. Use either IP address or
# logical name. (If you use a logical name be sure the name and its
# corresponding ip address appear in the sys:etc\hosts file.)
# By default, the local node sends traps at least to itself.

127.0.0.1      # send traps to the loopback address
123.123.34.56 # IP address of SunNet Manager system
```

23.2.3 Example Forwarded Trap

Following is a typical trap report which has been generated by SunNet Manager after receiving a forwarded trap from NMS. In this particular example, a supervisor has established a login connection with the server STEAM.

```
Mon Sep 20 15:08:56 1993 [ steam ] : Trap:
sequence=1
receive-time=Mon Sep 20 15:08:56 1993
version=0
community=PUBLIC
enterprise=Novell.mibDoc.nwalarm-mib
source-time=87:09:54.30
trap-type=loginConnection
serverName=STEAM
trapTime=748555591
userName=SUPERVISOR
connectionNumber=1
```

This chapter discusses the following topics:

- Set Tool windows
- Set information list
- Invoking Set Tool from the command line

Use Set Tool to see the details of attribute values and to change the values. You invoke the Set Tool with the Set Request option of the element's Glyph menu.

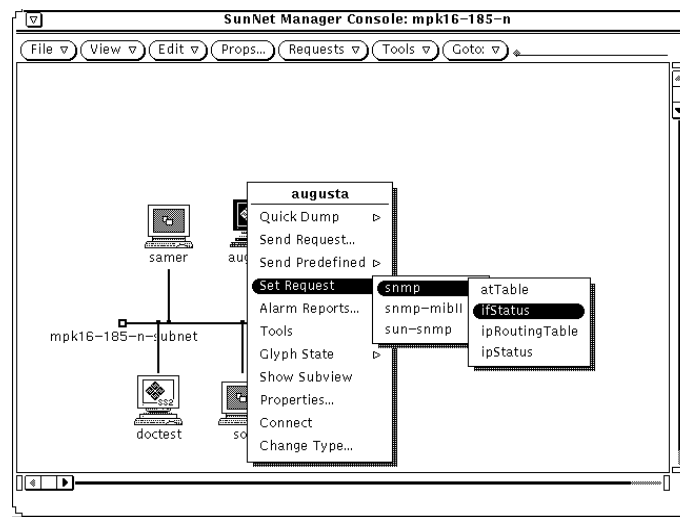


Figure 24-1 Invoking Set Tool

When Set Tool is invoked, a pop-up window is displayed as shown in Figure 24-2.

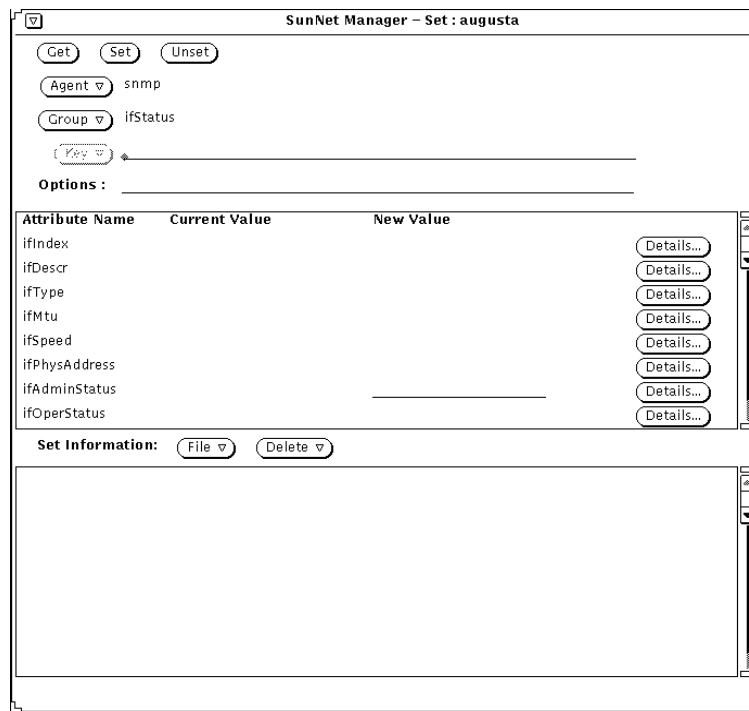


Figure 24-2 Set Tool Window

24.1 Set Tool Window

This section briefly describes the functions of the Set Tool window. Using the window functions to perform Get and Set requests are described later.

The system to be affected is shown to the right of Set in the title bar of the window. The Set Tool window is divided into three portions:

- The top portion is a control panel that contains control buttons and text item fields. The control panel fields determine what is displayed in the middle portion of the window. Control panel buttons and fields are described in more detail below.

- The middle portion is an attribute list that displays a group of attributes and their values. The attributes that are displayed in the attribute list are specified in the control panel. Starting with version 2.3, you can click the *Details* button to see details of an attribute. More information is in the section below, “Attribute Details,” provides more information.
- The bottom portion is a Set Information list. The Set Information list displays any new values that are entered into the attribute list. The Set Information list allows you to collect changes to multiple attributes (from one or more agents or groups) to be set in one operation.

24.1.1 Control panel buttons

The control panel buttons have the following functions:

- Get displays the current values for a group of attributes, or in the case of a table, a row of attributes.
- Set sends a Set request with the current set information.
- Unset undoes, if possible, the last successfully completed Set request.

The text item fields in the control panel allow you to specify the characteristics of the attributes displayed in the attribute list. These fields have the following functions:

- The Agent and Group fields are menus that allow you to select other settable agents and groups for the same system. To choose a different agent or group, press MENU on the abbreviated menu button.
- The Key field is only displayed if the group is a table and allows you to specify a key to identify a row (an instance) of attributes in the table. If you know the key of the row you want, you can simply type in the value on the line. If you do not specify a key value when making a get request, the first key in the table is shown and the Key menu becomes available with the abbreviated menu button. The Key menu consists of all the keys of the rows in a table at the moment that you make the Get request.
- The Options field allows you to specify the community name that the SNMP proxy agent should use when making a Get or Set request. The community name specified in this field overrides the community name that is defined in the Properties window for the target element.

24.1.2 Attribute Details

Click the *Details* button to see a window similar to the one in Figure 24-3.

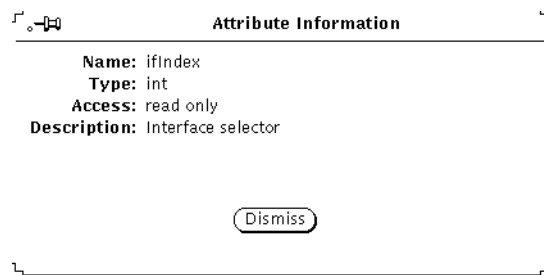


Figure 24-3 Set Tool Attributes Window

This is a read-only window. The fields have the following meanings:

- Name: name of the attribute
- Type: type of the attribute; if the attribute is an enumeration, the valid values are listed
- Access: how the attribute may be accessed
- Description: what the attribute means

24.2 Set Information List

The Set Information list allows you to set the values of multiple attributes—not necessarily from the same group or agent—in a single operation of the Set Tool. As individual new values are entered, they are appended to the Set Information list. When you select the Set button, the Set Tool makes the appropriate number of agent requests to change the selected values.

To change the value of an attribute displayed in the Set Information list, double-click the SELECT button on the desired attribute. The control panel and the attribute list are updated with the information about the selected attribute. You can then change the New Value column. To delete a line from the Set Information list, first click SELECT on the line to highlight it, then press MENU on the Delete abbreviated menu button and release on Delete Selection. You

can also delete multiple lines at a time by selecting the lines and then using the Delete Selection option. The Delete All option from the Delete menu clears the entire Set Information list.

The File button menu allows you to store the current Set Information list to a specified file, or load previously stored Set Information data. The format of the file should be exactly the same as the contents of the Set Information list. Each line in the file (and in the Set Information list) contains the set information for a single attribute. It has the following format:

```
<agent>/<group>/<attribute>(<key>) New Value = <newvalue>
```

where:

<agent>
is the agent name of the attribute to be set.

<group>
is the group name of the attribute to be set.

<attribute>
is the attribute name to be set.

(<key>)
is the key of the attribute to be set, if the attribute is in a row in a table.

<newvalue>
is the new value for the attribute to be set to.

The above variables are delimited by the slash (/) character. One or more blanks must precede the keyword `New Value`.

If the file to be loaded has any agent or group data which does not apply to the current target system, the data is not loaded. Data loaded from a file is appended to any entries in the Set Information list.

While a Get or Set request is being performed, a busy pointer is displayed and the title bar of the Set Tool window is dimmed. The Set Tool window does not accept any input until a response to the request is received from the agent. If the request is taking an inordinate amount of time, perhaps due to an agent error, use the Quit option of the window menu to exit the Set Tool.

24.3 Invoking Set Tool from the Command Line

The Set Tool is implemented as a separate process and can be invoked from the command line (whether or not the SunNet Manager Console is running). The `snm_set` command line tool attempts to use whatever runtime database is available on the machine where the command is entered. If the SunNet Manager Console is running (or was running at some time), the Console's runtime database is used. If no database is found, `snm_set` returns an error message. The format for invoking the Set Tool, is:

```
host% <tools-path>/snm_set -t <target_system> -a <default_agent_name> -g <default_group_name>
```

where:

`<tools-path>` is:

- `/usr/snm/bin` for Solaris 1.1 installations
- `/opt/SUNWconn/snm/bin` for Solaris 2.x installations.

`-t <target_system>`

is the name of the system you want to change attribute values on.

Note – If the name `<target_system>` contains spaces, then it must be surrounded by quote marks — for example, “jaks machine”.

`-a <default_agent_name>`

is the name of the default agent you want to change attribute values for.

`-g <default_group_name>`

is the name of the default group you want to change attribute values for.

Glossary

This is a glossary of terms used in the SunNet Manager documentation. Wherever possible, terms have meanings similar to standard usage within the network management community. Some definitions have been slightly modified to better illustrate unique SunNet Manager concepts.

activity daemon

Process that keeps a list of active agent requests that have been sent from the host where the daemon is running. The daemon periodically queries the agents to make sure they are still servicing their requests.

agent

A process, usually corresponding to a particular managed object, that carries out requests from a manager.

agent schema

A formatted description of the groups and attributes available from an agent, with an enumerated list of possible agent errors. This description enables manager processes to make requests for specific groups and attributes.

agent services

The function library that allows agents and managers to communicate, as viewed by the agent.

API

Application Programming Interface. The function library interface an application can call to perform a particular service.

attribute	A named data item corresponding to a property of a managed object.
attribute name	The name of an attribute as defined by the agent schema.
bus	A non point-to-point network type. Used by the MDB to represent a network link to which many other elements may connect.
cluster	A collection of records in the MDB which together represent an element. Cluster records are defined in instance files. Also called cluster record or instance record.
component	A type of network element corresponding to devices such as computers, operating systems, gateways, etc.
connection	A point-to-point network link connecting two other elements. Connections come in two types: regular and simple. Regular connections are full-fledged elements that can have properties, can be managed by agents, etc. Simple connections are a user-convenience feature that have a graphical representation but no database representation. (They don't have properties, can't be managed by agents, etc.)
Console	The window-based manager application that comes with SunNet Manager.
control	The ability to modify attribute values to change the characteristics of managed objects.
data report	Attributes routinely reported for analysis, as opposed to an event report.
decay to blue	The decay-to-blue feature pertains to events and traps that have reached a threshold and then fallen below it, or that have had an event or trap stop being reported. In response to these conditions, the glyph for the affected element turns blue.

discover function

Using this function, from the machine on which the SNM Console is running, the Discover Tool finds hosts, routers, networks, and Simple Network Management Protocol (SNMP) devices reachable from the Console machine.

dispatch function

Agent routine that calls an appropriate agent function to service a request.

element

An individual object subject to management. Examples: "the machine called mgrhost," "ethernet interface le0," "building 14," and so on.

element type

A structural definition of a particular kind of element. Examples: `component.sun3`, `bus.ethernet`, etc.

element category

A family of element types. The SunNet Manager Console defines four element categories: bus, component, connection and view.

event

The occurrence of a particular change in the state of an attribute.

event dispatcher

Process that acts as a rendezvous for event reports. Management applications can ask this daemon to send them selected copies of reports based on particular criteria.

event report

The notification an agent sends when an event is detected on a managed object.

gateway

A computer that interconnects two networks and routes packets from one to the other. A gateway has more than one network interface.

glyph

A pictorial representation. Similar to icon (except OPEN LOOK distinguishes between the two). The Console provides three forms of glyphs: the fixed-sized image (components, views and connections without both end elements in the current view), the variable length vector with grab handles (buses) and the variable length vector (connections with both end elements in the current view).

group	A collection of related attributes. Defined by the agent writer and specified in the agent schema.
instance	The data representing a particular element. As used by the MDB, definitions of instances are called instance records.
instance file	A collection of agent request and element instance records.
managed object	A particular resource (computer system, network interface, etc.) subject to management. In the Console this concept is called an element.
MDB	Management database. The collection of all managed object attribute definitions, along with data definitions specific to the SunNet Manager Console.
manager	The collection of programs or processes that work on behalf of a user to help manage a particular set of networks and devices. Usually sends requests to agents, accepts collected data from them, and displays the data for the user. Also called manager application or manager process.
manager services	The function library that allows agents and managers to communicate, as viewed by the manager.
manager station	The workstation where a human manages a particular set of networks and devices, usually running a manager application such <code>snm(1)</code> or <code>snm_cmd(1)</code> .
MIB	Management Information Base. The logical representation of network devices and their components as managed objects.
monitor function	Using this function, the Discover Tool compares the elements stored in the runtime database with the elements it finds at a specified interval or specified time. If new elements are detected, the monitor function stores the elements in

	a “holding area” view and records these elements in a log file. Through the same log file, the monitor function can also notify you if a previously discovered host is down or was down within a given period.
predefined request	Provides the user with an easy and quick method of making network management calls from a manager to an agent process, usually to collect and return specific attribute values.
proxy agent	An agent that manages on behalf of the manager. Used to manage objects not directly manageable due to different protocol suites or other incompatibilities. Sometimes shortened to proxy.
proxy system	The host where a proxy agent runs.
rendezvous	Process where an agent sends data and event reports. Also known as rendezvous process.
report	Answer (usually from an agent to a manager) containing attribute values or error messages. Sometimes called response.
request	Network management call from a manager to an agent process, usually to collect and return specific attribute values.
RPC	Remote Procedure Call.
schema	A description of element structures and instances. While technically every file (except icons and rasterfiles) that the Console loads into the runtime database is a schema, in practice, records that define the structure of elements are called structure definitions, and records that define the actual elements are called instance records. The remaining schema records, used by agents to define their capabilities, are called agent schema.
schema file	A UNIX file containing one or more agent schema. Also called agent schema file.

SNMP	Simple Network Management Protocol, the standard network management protocol used in TCP/IP networks.
SNMPv2	Simple Network Management Protocol Version 2, an enhanced version of the standard network management protocol used in TCP/IP networks.
stream	A logical grouping of reports created according to the name of the requesting agent, type of report, and the request timestamp.
structure file	A collection of agent and element structure definitions.
system	An instance of a component. Sometimes called host or device.
target system	The host where the managed object resides.
verify function	Agent routine that checks the validity of a request before the request is dispatched.
view	A collection of components in the Console's MDB. Views have members, called elements. Since views are elements, they may be members of other views. Elements may have multiple memberships. Memberships may be circular.

Index

Symbols

\$HOME/.SNMdefaults file, 13-8
\$HOME/.SNMpredefined, 15-21
\$HOME/.SNMpredefined file, 13-8
\$SNMHOME/defaults file, 17-30
.SNMdefaults file, 17-29
.SNMpredefined file, 13-8, 15-21
/etc/snm.conf file, 4-22
/tmp/snm_gr.rpcid. file, 21-3
/var/adm/snm directory, 3-7, 3-38, 13-5
/var/adm/snm/event.log file, 20-1
/var/adm/snm/snmp.traps, 19-13
/var/opt/snm directory, 13-5
/var/opt/snm/snmp.traps, 19-13
~/.SNMpredefined file, 15-21

Numerics

5.x directory, 13-6

A

absolute values

- graphing data sets, 21-6
- graphing time stamps, 21-6

active request state, 15-42

activity daemon, 13-1, 13-4, 13-7

definition of, Glossary-1
occurrences of, 13-5

activity.log file, 13-7

Add button

data report request, 15-30, 15-40

adding element types, 18-1

adding SNMP devices, 8-2

adding Tools for element type, 10-12

adding user commands, 18-2

agent, 13-1

definition of, Glossary-1
security, 11-4

agent configuration, 1-8

agent configuration file, 13-5

agent directory, 10-13

agent files, 13-6

agent header files, 13-6

agent record, 18-11

agent reports, 20-6

agent schema, 14-2

definition of, Glossary-1

agent schema files, 1-7, 18-2

agent services

definition of, Glossary-1

agents, 13-2

correlation between presence and
requests available, 5-24

- definition of, 1-2
- how to access information on, 2-9
- procedure to add, 10-12
- related to predefined requests, 4-13
- using Discover for custom ..., 22-29
- writing your own, 13-4
- agents directory, 13-6, 19-12
- agents for Sun products, 13-4
- agents, shipped with SNM
 - summary of, 2-8
- alarm report
 - definition of, 6-1
 - features of device-specific ..
 - window, 6-7
 - filtering, 6-7
 - finding agent-specific, 6-10, 16-6
 - printing device-specific, 6-11, 16-7
 - saving device-specific, 6-10, 16-7
 - sorting, 6-8, 16-4
 - viewing device-specific, 16-4
- Alarm Reports Summary window, 16-4
- alarms
 - forwarding NMS .. to SNM, 23-4
- Alarms Reports
 - viewing, 16-2
- Alias button, 14-29
- Alias name, 14-30
- aliases
 - use in managing multiple network
 - interfaces, 3-26
- Aliases List, 14-30
- alternate proxy, 15-15
- analyzing data reports
 - Results Browser, 4-22
 - Results Grapher, 4-29
- API
 - definition of, Glossary-1
- Apply button
 - Console properties, 17-4
 - data report request, 15-10
 - Request Builder template, 15-11
- apply button, 12-8
- ASCII database file
 - specifying in SNM command line, 3-5
- asynchronous event reports (traps), 19-7, 19-12
- attribute
 - data report request, 15-9, 15-28
 - definition of, Glossary-2
- attribute name
 - definition of, Glossary-2
- attribute values
 - changing, 8-18
 - data report request, 15-7, 15-14, 15-27, 15-33
 - event report request, 15-15, 15-34
- attributes, 1-2, 1-7, 14-1
- Audio Effect, 17-11
- Audio File, 17-11
- audio file, 15-19, 15-39
- audio signal for trap, 17-21
- audio signal options
 - event report request, 15-19, 15-38
 - play audio file, 15-19, 15-38
 - ring bell, 15-19, 15-38
- Automatic Management, 17-10
- automatic management, 17-7
 - audio effect, 17-11
 - audio file, 17-11
 - event priority, 17-11
 - glyph effect, 17-11
 - polling interval, 17-11
 - send mail, 17-12
 - send to program, 17-12
- Automatic Management category of
 - Console Properties, 17-7
- automatic node management, 17-7
 - enabling, 17-10
- available attributes
 - event report request, 15-16, 15-35
- awaiting activation request state, 15-42
- awaiting stop request state, 15-42

B

- background image

- adding to current view, 10-1
 - comparison of means of adding, 10-5
- background image - definition, 18-12
- background images - definition, 18-2
- banner
 - Console, 14-2
- BasicStart button, 3-7
- being activated request state, 15-42
- bin directory, 13-6
- binary files, 13-6
- blink glyph signal option, 15-18, 15-37
- blinking glyph
 - priority, 17-18
- Browser, 4-22, 14-3, 14-18
 - agent reports, 20-6
 - customizing, 20-15
 - file data format, 20-1
 - folders, 20-13
 - loading a file, 20-3
 - report streams, 20-5
 - sending data to Grapher, 20-11
 - starting, 20-2
- build
 - Predefined Data Request, 15-20
 - Predefined Event Request, 15-20
- bus
 - definition of, Glossary-2
- buses, 18-3
 - Ethernet LAN, 18-3
- buttons
 - Edit, 14-11
 - File, 14-6
 - Goto, 14-19
 - Props, 14-17
 - Tools, 14-18

C

- cause of events, 5-25
- Circular layout style
 - node frame percent, 12-33
- Change button
 - data report request, 15-30, 15-40
- changing an attribute value, 8-18
- changing element type, 14-31
- changing glyph state, 5-27
- changing SNMP attributes, 8-18
- circular layout style, 12-32 to 12-41
 - cluster by group ID, 12-39
 - cluster by IP address, 12-37
 - edge degree, 12-36
 - grouping options, 12-34
 - grouping window, 12-39
- clients discover IPX/SPX, 23-1
- clipboard, 14-12, 16-19
 - function in copying elements, 3-34
 - function in deleting elements, 3-36
 - function in moving elements, 3-32
- cluster
 - definition of, Glossary-2
- cluster record, 18-15
- cluster records, 18-9
- color, 12-13
 - element, 14-29
 - priority, 17-18
 - specifying the .. of an element, 3-20
- color of graph, 21-6
- colormap entries, 17-5
- colors
 - specifying .. for discovered components, 22-26
- command-line interface, 13-12
- community string, 24-4
- completion of request, 15-7, 15-14, 15-27, 15-33
- component
 - changing element type, 14-31
 - definition of, Glossary-2
- component element type definition, 18-6
- components, 18-3
 - printers, 18-3
 - routers, 18-3
 - workstations, 18-3
- conferring right-of-access, 11-4
- configuration, 1-8
 - Console, 17-1

- IP Discover, 22-6
- configuring colormap entries, 17-5
- configuring error effect, 17-23
- configuring error signals, 17-23
- configuring event effect, 17-17
- configuring Goto button operation, 17-27
- configuring horizontal scrolling, 17-6
- configuring icon fonts, 17-5
- configuring location of icon directories, 17-26
- configuring request timeout, 17-7
- configuring Restart field default, 17-6
- configuring schema file locations, 17-25
- configuring trap effects, 17-19
- configuring username display, 17-5
- configuring vertical scrolling, 17-6
- configuring window title, 17-5
- connect record, 18-11
- connecting elements, 3-32, 14-30
- connecting two glyphs, 14-31
- connection
 - definition of, Glossary-2
 - simple vs. manageable distinction, 22-11
- connection - definition, 18-11
- connection element type definition, 18-4
- connections, 18-3
 - distinction between manageable and simple ..., 3-33
 - leased internetwork link, 18-3
 - RS-232-C line, 18-3
- connections - definition, 18-2
- Console
 - definition of, Glossary-2
 - introduction to, 1-5
 - multiple instances, 14-2
 - quitting, 3-39
 - rules for control-area functions, 3-8
 - starting up, 3-4
 - user name, 14-2
- Console configuration, 1-8, 17-1, 17-29
- Console customization, 17-30
- Console lock file, 13-7
- Console operation
 - Edit button, 14-11
 - Glyph menu, 14-20
 - Goto button, 14-19
 - Props button, 14-17
 - Requests button, 14-17
 - Tools button, 14-18
- console operation
 - File button, 14-6
- Console Properties, 17-2
 - Category menu, 17-3
 - Events and Traps, 17-16
 - fonts for glyph labels, 17-5
 - Locations, 17-25
 - Miscellaneous, 17-26
 - Requests, 17-6
 - Windows
 - Display Username in Title, 17-5
 - Horizontal Scrolling, 17-6
 - Icon Font, 17-5
 - Maximum Colors, 17-5
 - Vertical Scrolling, 17-6
 - Window Title, 17-5
- Console properties, 14-17
 - Apply button, 17-4
 - Automatic Management, 17-7
 - effects, 17-4
 - Errors, 17-22
 - Reset button, 17-4
- Console properties categories, 17-4
- Console views, 3-16
- control
 - definition of, Glossary-2
- control panel of Set Tool, 24-4
- Copy
 - editing, 14-12
- copying elements, 3-34, 14-12, 14-13
- copying requests, 4-14
- count, 15-6, 15-13, 15-25, 15-32
- Create
 - editing, 14-13
- creating a background canvas, 10-1

creating an SNMP element, 8-5
 creating element types, 10-6
 creating elements, 3-18, 14-13
 creating glyphs for element types, 10-9
 creating SNMP elements, 8-5
 critical nodes
 using event requests to monitor, 5-8
 custom colors
 configuring, 17-29
 customizing Console, 17-30
 Cut
 editing, 14-12
 cutting elements, 14-12

D

daemon configuration, 1-8
 daemon configuration file, 13-5
 daemon for, 19-27
 daemons, 13-1, 13-4
 data files, 4-22
 data log
 data report request, 15-9, 15-29
 examining, 16-9
 data report, 1-6, 14-3
 definition of, Glossary-2
 indicator, 16-11
 security, 11-1
 Strip Chart, 16-12
 Data Report entry format, 16-10
 data report request
 Add button, 15-30, 15-40
 Agent Schema, 15-25
 Apply button, 15-10
 attribute values, 15-7, 15-14, 15-27, 15-33
 Change button, 15-30, 15-40
 count, 15-6, 15-13, 15-25, 15-32
 data log, 15-9, 15-29
 defer reports, 15-6, 15-13, 15-26, 15-33
 Define button, 15-30, 15-40
 Delete button, 15-10, 15-30, 15-40
 disposition of request, 15-7, 15-14, 15-27, 15-33
 file name, destination of ASCII text, 15-8, 15-27
 Grapher Tool, 15-10, 15-29
 group/table, 15-25
 Hold button, 15-11
 indicator, 15-9, 15-29
 interval, 15-6, 15-13, 15-32
 key, tabular information, 15-6, 15-13, 15-25, 15-32
 naming a request, 15-5, 15-12, 15-24
 option string, 15-7, 15-14, 15-26, 15-32
 polling interval, 15-25
 proxy system, 15-6, 15-12, 15-31
 Request Name, 15-24
 Reset button, 15-10, 15-11, 15-30, 15-40
 restart, 15-6, 15-13, 15-26, 15-32
 specify a particular attribute, 15-9, 15-28
 Start button, 15-11
 strip chart, 15-10, 15-29
 Undefine button, 15-30, 15-40
 data reporting, 14-3
 data reports
 maximum number, 17-27
 Data Reports window, 4-16, 16-9
 maximum number, 17-27
 data request
 analyzing results, 4-22
 how to send, 4-5
 modifying, 4-35
 Quick Dump, 4-3
 viewing results, 4-15
 data request - definition, 18-15
 Data Request template, 15-5
 data requests - definition, 18-2
 data sets
 graphing, 21-6
 order in graphing, 21-7
 database, 14-2
 changes, 18-1
 definitions, 18-1

database API lock file, 13-7
 database change traps, 17-22
 database directory, 13-5, 13-7
 database, create Sun Net Manager
 database, 12-4
 databases
 managing duplicate, 18-23
 databases, managing duplicate, 3-6
 dataRequest record, 18-15
 decay feature
 use with manageable
 connections, 9-9
 Default file locations, summary of, 13-10
 default request timeout, 17-7
 default value of Restart field, 17-6
 defaults button, 12-8
 defer reports, 15-6, 15-13, 15-26, 15-33
 sending, 15-47
 Define button
 data report request, 15-30, 15-40
 definition
 glyph, 18-6
 Delete
 editing, 14-13
 Delete button
 data report request, 15-10, 15-30,
 15-40
 deleting elements, 3-36, 14-13
 dim glyph signal option, 15-18, 15-37
 dimmed glyph
 priority, 17-18
 directories
 log file, 13-5
 directories created at installation, 13-5
 disabling glyph state propagation, 5-30
 Discover, 14-18
 discover
 configuring IP-Discover, 3-11
 Discover configuration window, 22-4
 discover function
 IP discover, 22-1
 Discover Properties
 ICMP Retries, 22-7
 ICMP Timeout, 22-17
 Verbose Mode, 22-18
 Discover Tool, 14-3
 distinctions among element
 types, 3-31
 invoking, 22-2
 Monitor
 Cycle Time, 22-17
 Holding Area Viewname, 22-17
 Ignore Host File Name, 22-18
 Log Filename, 22-17
 Log History, 22-17
 Mail Logfile To, 22-17
 Response Timeout, 22-17
 Start Date, 22-18
 Start Time, 22-18
 Stop Date, 22-18
 Stop Time, 22-18
 monitor function, 3-9, 22-1, 22-15
 used to create manageable
 connections, 9-10
 using with large subnetworks, 22-24
 discover, invoking IP-Discover, 3-9
 discover, viewing elements, 3-10
 discover.conf file
 location and description of, 22-25
 Disk Space Requirements, 3-6
 Display username, 17-5
 displaying empty views, 17-27
 displaying request properties, 15-44
 disposition of request upon
 completion, 15-7, 15-14, 15-27,
 15-33
 Drop All popup menu, 16-17
 duplicate databases
 managing, 18-23
 duplicate databases, managing, 3-6

E

Edit button, 14-11
 Edit menu
 Copy, 14-12

- Create, 14-13
- cut, 14-12
- Delete, 14-13
- Paste, 14-12
- Select all, 14-12
- effect of event, 17-19
- effect propagation, 17-17
- effects of Console properties changes, 17-4
- element
 - definition of, Glossary-3
- element category
 - definition of, Glossary-3
- element Glyph menu
 - Change Type, 14-31
 - Connect, 14-30
 - Properties, 14-26
 - Quick Dump, 14-21
 - Send Predefined, 14-22
 - Send Request, 14-22
 - Set request, 14-22
 - Show Subview, 14-25
 - Tools, 14-23
- element glyph state
 - changing, 5-27
 - propagation, 5-30
- element instance - definition, 18-2, 18-9
- element instance definition
 - agent record, 18-11
 - connect record, 18-11
 - glyphState record, 18-11
 - instance type record, 18-10
 - membership record, 18-10
 - proxy record, 18-11
- element instance names, 18-10
- element instances, 1-7
- element properties, 3-28
- element Properties window, 3-19
- element schema
 - defining an icon mask, 10-9
- element schema file, 1-6
- element schema files, 18-2
- element type
 - adding Tools for, 10-12
 - changing, 14-31
 - changing component, 3-29
 - creating glyphs for, 10-9
 - definition of, Glossary-3
 - modifying Tools for, 10-12
- element type - definition, 18-2, 18-3
- element type categories, 18-3
- element type definition
 - connections, 18-4
 - Glyph_State, 18-4
 - Label, 18-5
 - Name field, 18-4
 - user command, 18-8
- element type definition file, 13-6
- element type definition record, 18-4
- element type definitions, 1-6
- element types, 18-1
 - creating, 10-6
 - restrictions for changing, 3-30
- elementCommand instance, 18-8
- elementGlyph, 10-10
- elementGlyph instance, 18-6
- elements
 - connecting, 3-32
 - copying, 3-34, 14-12, 14-13
 - creating, 3-18, 14-13
 - creating elements within, 3-24
 - creating SNMP elements, 8-5
 - creating with the editor, 3-18
 - cutting, 14-12
 - deleting, 3-36, 14-13
 - finding, 3-17, 3-26
 - glyphs for multiple types, 3-21
 - modifying properties of, 3-28
 - moving between views, 3-32
 - pasting, 14-12
 - selecting, 14-12
- elements.schema file, 1-6, 13-6, 18-3, 21-3
- empty views, 17-27
- enabling automatic node management, 17-10
- enabling propagation of effect, 17-17

- enterprise object identifier, 19-15
- enterprise object identifier (OID), 19-12
- enterprise-specific trap, 19-15
- enterprise-specific traps, 19-12, 19-14
- environment variables, 13-9
 - HELPPATH, 13-9
 - MANPATH, 13-9
 - SNM_NAME, 13-7
 - SNMDDIR, 3-7, 13-5, 13-7
 - SNMDISCOVERMAP, 13-10
 - SNMHOME, 13-5
 - SNMLINKMAP, 13-10
- error effect, 17-23, 17-24
 - propagating, 17-24
- error messages, 6-11
- Error Reports
 - maximum number, 17-27
 - viewing, 16-16
- Error Reports window, 6-12
 - maximum number, 17-27
- error signals, 17-23
- Errors category of Console
 - Properties, 17-22
- event, 1-6
 - definition of, Glossary-3
 - detecting presence of, 5-24
 - specifying, 5-2
- Event Dispatcher, 19-16
- event dispatcher, 13-1, 13-4, 13-7
 - definition of, Glossary-3
 - occurrences of, 13-5
- event effect, 17-19
- event glyph changes, 5-27
- event notification
 - audio signal options, 15-19, 15-38
 - mail options, 15-20, 15-39
 - program options, 15-20, 15-39
- Event Priority, 17-11
- event report
 - definition of, Glossary-3
 - polling interval
 - event report
 - count with start/stop, 15-34
- security, 11-1
- event report request, 5-2
 - agent schema, 15-25
 - attribute values, 5-23, 15-15
 - audio effect, 15-19, 15-38
 - available attributes, 15-16, 15-35
 - count, 15-6, 15-13, 15-25, 15-32
 - count attribute, 15-34
 - defer reports, 15-6, 15-13, 15-26, 15-33
 - disposition of request, 15-7, 15-14, 15-27, 15-33
 - file name, destination of ASCII
 - text, 15-8, 15-27
 - glyph effect, 15-18, 15-37
 - group/table, 15-25
 - interval, 15-6, 15-13, 15-32
 - key, tabular information, 15-6, 15-13, 15-25, 15-32
 - mail options, 15-20, 15-39
 - naming a request, 15-5, 15-12, 15-24
 - option string, 15-7, 15-14, 15-26, 15-32
 - polling interval, 15-25
 - priority, 15-18, 15-37
 - program options, 15-20, 15-39
 - proxy system, 15-6, 15-12, 15-31
 - relation, 15-16, 15-35
 - Request Name, 15-24
 - restart, 15-6, 15-13, 15-26, 15-32
 - threshold, 15-18, 15-37
 - threshold conditions, 15-17, 15-36
 - threshold value, 15-18, 15-37
 - visual signal options, 15-18, 15-37
- event report requests
 - automatic, 17-7
- event reporting, 14-3
- event reports
 - maximum number, 17-28
 - modifying maximum number, 5-27
 - obtaining, 5-25
- event request
 - modifying, 4-35
 - properties sheet, 5-14
 - sending, 5-4
- event request - definition, 18-19
- event requests - definition, 18-2

- event.log file, 4-22, 13-7
- event/trap Log
 - interpreting traps, 19-23
- event/trap log
 - examining, 16-15
- Event/Trap Reports
 - viewing, 16-14
- Event/Trap Reports window, 5-25, 6-15, 16-14, 19-23
 - maximum number, 17-28
- event-based requests
 - scheduling, 5-2
- eventRequest record, 18-15
- Events
 - clearing with Drop All option, 16-17
- events
 - checking cause, 5-25
 - decay to blue, 15-19, 15-38, 17-20
 - SNM response to, 5-2
- Events and Traps category of Console Properties, 17-16
- events.ind file, 13-7
- events.rec file, 13-7
- examining data reports log, 16-9
- examining event/trap log, 16-15
- example database, 13-7
- example.db file, 13-7

F

- fatal errors, 17-23
- File button, 14-6
- file formats, 20-1
- File locations, summary of, 13-10
- files created at installation, 13-5
- filters
 - setting host specific, 8-10, 19-7
- finding elements, 3-17, 3-26, 16-18
- folders, 20-13
 - creating, 20-13
 - deleting, 20-14
 - emptying, 20-14
- fonts

- changing label, 3-24
- forking programs for element creation or modification, 17-30

G

- gateway
 - definition of, Glossary-3
- Get button
 - Set Tool, 24-4
- get requests, 8-13
- glyph
 - definition of, Glossary-3
 - returning to normal color, 15-19, 15-38, 17-20
- glyph definition, 18-6
- glyph directory, 10-13
- Glyph Effect, 17-11
- glyph effect
 - customizing color of effect of traps on defaults for, 17-19
 - event report request, 15-18, 15-37
- Glyph Menu
 - Quick Dump, 14-21
- Glyph menu, 14-20
 - Change Type, 14-31
 - Connect, 14-30
 - Properties, 14-26
 - Send Predefined, 14-22
 - Send Request, 14-22
 - Set Request, 24-1
 - Set request, 14-22
 - Show Subview, 14-25
 - Tools, 14-23
- glyph menu
 - Glyph State, 17-18
- glyph state
 - blinking, 17-18
 - changing, 5-27, 17-18
 - color, 17-18
 - dimmed, 17-18
 - priority, 17-18
 - propagation, 5-30

-
- resetting, 17-18
 - view, 17-18
 - Glyph State option, 5-27, 17-18
 - Glyph_State, 18-4
 - glyphs, 3-1, 18-2
 - attributing traps to, 19-14
 - positioning, 14-17
 - procedure to add, 10-12
 - request, 15-45
 - use of for multiple elements, 3-20, 14-16
 - glyphs for element types, 10-9
 - glyphState record, 18-11
 - Goto button, 3-17, 14-19, 17-27
 - graph
 - data report request, 15-10, 15-29
 - Graph Properties window, 21-4
 - Graph Tool, 4-17, 16-14
 - Grapher, 4-15, 4-29, 14-3, 14-18
 - b option, 21-3
 - displaying graphs, 21-8
 - invoking from Browser, 4-42
 - invoking from the command
 - line, 21-2
 - Merge button, 21-4
 - properties, 21-4
 - Properties button, 21-4
 - Remove button, 21-4
 - starting, 21-2
 - temporary RPC program
 - number, 21-3
 - View button, 21-4
 - Grapher Window, 21-3
 - graphical editing, 14-11
 - graphical editor
 - copying elements, 14-12
 - creating elements, 14-13
 - cutting elements, 14-12
 - deleting elements, 14-13
 - pasting elements, 14-12
 - selecting all elements, 14-12
 - graphs
 - background color, 21-3, 21-9
 - base color, 21-6
 - data scale, 21-6, 21-9
 - data set order, 21-7
 - dimension specification, 21-7
 - displaying, 21-8
 - drawing style, 21-7
 - fill, 21-7
 - frame color, 21-6
 - hide, 21-7
 - line color, 21-8
 - line style, 21-8
 - merging, 21-11
 - name, 21-6
 - plotting values, 21-8
 - printing, 4-31
 - replotting, 21-10
 - show, 21-7
 - start and end times, 21-6
 - time scale, 21-6, 21-10
 - viewing angles, 21-10
 - zooming, 21-10
 - group
 - definition of, Glossary-4
 - grouping window
 - group filter, 12-40
 - refresh list, 12-40
 - sort by, 12-40
 - uingroup all, 12-41
- ## H
- header files, 13-6
 - HeadStart button, 3-7
 - help file, 13-9
 - HELPPATH environment variable, 13-9
 - hiding graphs, 21-7
 - hierarchical layout style, 12-20 to ??, 12-22
 - to 12-31
 - bus size, 12-30
 - level alignment, 12-24
 - level frame percent, 12-24
 - level orientation, 12-21, 12-23
 - minimum slope, 12-28
 - node frame percent, 12-26
 - stagger end nodes, 12-32
 - Hold button

- data report request, 15-11
- Home view, 3-16
- Horizontal Scrolling, 17-6
- hostperf proxy agent, 3-20

I

- icon directories, 17-26
- icon directory, 10-13
- icon files, 13-6
- Icon Font, 17-5
- icon mask, 10-9
- icons directory, 13-6
- include directory, 13-6
- incoming data
 - viewing, 4-15
- Indicator, 4-15
 - data report, 16-11
- indicator
 - data report request, 15-9, 15-29
- Indicators, 4-18
- installation directory, 13-5
- instance
 - definition of, Glossary-4
- instance file
 - definition of, Glossary-4
- instance files, 18-2
- instance type record, 18-10
- instances of elements, 1-7
- interpreting traps, 19-23
- interval
 - request properties, 15-6, 15-13, 15-32
- invoking Discover Tool, 22-2
 - Console Tools window, 22-2
 - SunOS command line, 22-2
- IP address
 - using as name of an element, 3-19
- IP Discover
 - configuration, 22-6
- IP Discover properties
 - Add Object Connections, 22-11
 - Add Object Coordinates, 22-11

- Add To All Views, 22-10
- Default Proxy, 22-12
- Gateway File, 22-9
- ICMP Timeout, 22-7
- Maximum Hops, 22-6
- Net Name/Number, 22-6
- Netmask, 22-6
- Objects to Discover, 22-12
- Search Method, 22-8
- SNMP Community, 22-7
- SNMP Retries, 22-8
- SNMP Timeout, 22-8
- Verbose Mode, 22-10
- Viewname, 22-10

K

- key, 15-6, 15-8, 15-13, 15-25, 15-27, 15-32, 24-4
- killing a request, 15-46

L

- Label, 18-5
- launch
 - Predefined data requests, 14-20
 - Quick Dump requests, 14-20
 - standard requests, 14-20
- layout
 - all views check box, 12-7
 - style, 12-7
 - style menu, 12-7
- layout option, 12-6
- layout style
 - choosing, 12-8
- lib directory, 13-6
- library files, 13-6
- line color of graphs, 21-8
- line graphs, 21-7
- line style of graphs, 21-8
- link creation
 - using Console's editor, 9-2
- link management
 - description of, 9-1

- interaction with Discover Tool, 22-11
- link naming convention, 9-12
- linkmap file, 13-9, 13-10, 22-11
 - use in link management, 9-5
- list of supplied Predefined Data requests, 15-22
- list of supplied Predefined Event requests, 15-23
- Load menu
 - Predefined Requests option, 14-9, 14-10
- Load option
 - ASCII-format files only, 14-9
 - Predefined Requests, 14-10
- Load window, 17-26
- localhost, 14-28
- locations
 - icons directories, 17-26
 - schema files, 17-25
- Locations category of Console Properties, 17-25
- lock file, 13-7
- log file directory, 13-5
- log file locations, 13-6
- log files
 - directory, 13-6
 - installation, 13-6

M

- mail options
 - event report request, 15-20, 15-39
- mailing trap report, 17-22
- main window controls, 12-7
- man directory, 13-6, 13-9
- man pages directory, 13-6
- manageable connection
 - properties of, 9-3
- managed object
 - definition of, Glossary-4
- management application, 1-2
- management database, 1-6, 14-2
 - changes, 18-1
 - definitions, 18-1
- management station, 1-2
- manager
 - definition of, Glossary-4
- manager services
 - definition of, Glossary-4
- manager station
 - definition of, Glossary-4
- manager-to-manager capability, 19-28
- managing SNMP devices, 8-1
- MANPATH environment variable, 13-10
- Maximum Colors, 17-5
- maximum data reports, 17-27
- maximum error reports, 17-27
- maximum event reports, 17-28
- maximum trap reports, 17-28
- MDB, 14-1
 - definition of, Glossary-4
- MDB files, 14-2
- membership record, 18-10
 - positioning coordinates, 18-10
 - view name, 18-10
- menu
 - Edit, 14-11
 - file, 14-6
 - Glyph, 14-20
 - Goto, 14-19
 - Requests, 15-1
 - Tools, 14-18
- merging graphs, 21-11
- MIB
 - definition of, Glossary-4
- Miscellaneous category of Console Properties, 17-26
- modifying a data request, 4-35
- modifying a request, 15-46
- modifying an event request, 4-35
- modifying element type glyphs, 18-2
- modifying element types, 18-1
- modifying Tools for element type, 10-12
- modifying user commands, 18-2
- Monitor

- using to update management database, 22-19
- monitor function, 22-1
 - time parameters for, 22-19
- monitor.log file, 13-8
- moving elements, 3-32
- multiple Console instances, 14-2

N

- na.activity, 13-5
- na.event, 13-5
- na.hostif, 13-2
- na.logger agent, 13-8
- na.snmp proxy agent, 19-2
- na.snmp-trap daemon, 19-12
- name
 - data report request, 15-5, 15-12, 15-24
- Name field
 - in element property sheet, 3-19
- Name field of element type
 - definition, 18-4
- name of graph, 21-6
- navigating networks, 12-1
- nc.ind file, 13-7
- nc.rec file, 13-7
- netmgt_db.h file, 18-3
- network
 - print views, 12-14
- network layout assistant, 12-1
- New Folder, 20-13
- nla, 12-1
 - what it does, 12-1
 - what it does not do, 12-2
- NMS alarms
 - forwarding to SNM, 23-4
- nodes, discover IPX/SPX, 23-1
- Novell's NetWare Management System
 - forwarding alarms to SNM, 23-4
- number of requests, 15-42
- NXIS and Sun Net Manmager, 23-1

O

- Object Identifier Database (OID)
 - displaying, 19-26
- Object1 field of element type
 - definition, 18-4
- Object2 field of element type
 - definition, 18-4
- occurrences of activity daemon, 13-5
- occurrences of event dispatcher, 13-5
- OID
 - mapping to element types in Discover, 22-26
 - see Object Identifier Database, 19-26
- opening Console window, 17-17
- OpenWindows, 14-2
- option string, 15-7, 15-14, 15-26, 15-32, 24-4
- Options field
 - Set Tool, 24-4
- order of data sets
 - graphing, 21-7
- overview window, 12-10 to 12-14
 - color, 12-13
 - process all traps check box, 12-12
 - process change traps check box, 12-12
 - refresh contents button, 12-12
 - shapes, 12-12
 - troubleshooting, 12-13

P

- Paste
 - editing, 14-12
- pasting elements, 14-12
- PATH environment variable, 13-9
 - adding SNM to, 3-2
- per-product configuration, 17-30
- per-user configuration, 17-29
- ping proxy agent, 3-20
- planning
 - for network management, 2-1
- play audio file signal option, 15-19, 15-38
- plotting values in graphs, 21-8

Polling Interval, 17-11
 polling interval
 request properties, 15-25
 position of glyphs, 14-17
 positioning coordinates in membership
 record, 18-10
 predefined data and event requests, 13-8
 predefined data requests
 sending from glyph menu, 4-11
 summary of, 4-37
 predefined event report requests, 17-7
 predefined event requests
 summary of, 5-8
 predefined request, Glossary-5
 predefined requests, 13-8, 14-17, 15-1,
 15-20
 depending on agents present, 4-13
 print views of network, 12-14
 printers
 how to manage, 7-1
 printing a graph, 4-31
 printing Console screens, 14-18
 priority
 blinking, 17-18
 color, 17-18
 dimmed, 17-18
 event report request, 15-18, 15-37
 glyph state, 17-18
 priority sets color signal option, 15-19,
 15-38
 program directories, 13-5
 program files, 13-5
 program options
 event report request, 15-20, 15-39
 propagating error effect, 17-24
 propagation
 blinking priority, 17-18
 color priority, 17-18
 dimmed priority, 17-18
 glyph state priority, 17-18
 propagation of event effect, 17-17
 propagation of glyph state, 5-30
 propagation of trap effect, 17-21
 properties
 Console, 14-17
 element glyph, 14-26
 finding .. of a request, 4-13
 properties of a request, 4-35
 properties of an element, 3-19, 3-28
 properties window
 agent schema list, 14-28
 color, 14-29
 component data, 14-26
 Props button, 14-17
 protocol operations, 19-28
 proxy agent, 1-2, 1-4, 13-1
 default location, 14-28
 definition of, Glossary-5
 proxy agents, 13-2
 proxy feature
 ability to access hostperf and ping
 information, 5-24
 proxy record, 18-11
 proxy system
 definition of, Glossary-5
 request properties, 15-6, 15-12, 15-31
 pseudo-device
 element definition for, 19-20
 pseudo-devices, 19-17

Q

-q option
 to snm command, 3-7
 Quick Dump
 how to make .. request, 4-2
 Quick Dump request, 4-3
 Quick Start window
 how to suppress, 3-7
 quit Console verification, 17-27
 quitting a Console session, 3-39

R

read community, 8-6

read-security, 11-1
 receiving SNMP traps, 8-8
 record
 element type definition, 18-4
 redirecting SNMP requests, 17-31
 relation
 event report request, 15-16, 15-35
 relative values
 graphing data sets, 21-6
 graphing time stamps, 21-6
 rendezvous
 definition of, Glossary-5
 rendezvous process, 13-4
 replotting graphs, 21-10
 report
 definition of, Glossary-5
 Report menu, 20-8
 report streams, 20-5
 request
 definition of, Glossary-5
 Request Builder template
 Apply button, 15-11
 request glyph, 4-32, 15-45
 request name, 15-42
 request properties, 4-35
 request properties window, 15-44
 request restart interval, 17-7
 request state, 15-42
 request status, 4-33
 request timeout, 17-7
 request timestamp, 15-42
 request type, 15-41
 Request Viewer window
 Props button, 15-44
 selecting requests, 15-7, 15-14, 15-33
 Sort menu, 15-44
 request.log file, 13-8
 requests, 14-3
 copying, 4-14
 creating your own, 4-9
 numbers of, 15-42
 prioritizing, 4-5
 scheduling event-based, 5-2
 scheduling example, 5-23
 sending through Console Requests
 Menu, 4-6
 states of active or held, 4-34
 Requests button, 14-17
 Requests category of Console
 Properties, 17-6
 Requests menu, 14-17
 Create Predefined, 15-20
 Quick Dump, 15-2
 Send Request, 15-4
 Summary, 15-40
 Requests Summary window
 use in finding a request, 4-13
 Requests viewer, 4-33
 requests, definition of, 18-14
 Reset button
 data report request, 15-10, 15-11,
 15-30, 15-40
 reset button, 12-8
 Rest button
 Console properties, 17-4
 Restart, 17-6
 restart interval, 17-7
 restart requests, 13-8, 15-6, 15-13, 15-26,
 15-32
 restarting failed requests, 17-7
 Results Browser, 1-6, 4-22, 14-3, 14-18
 agent reports, 20-6
 customizing, 20-15
 folders, 20-13
 loading a file, 20-3
 report streams, 20-5
 using, 4-41
 Results Grapher, 1-6, 4-15, 4-22, 4-29, 14-3,
 14-18, 16-14
 -b option, 21-3
 displaying graphs, 21-8
 invoking from the command
 line, 21-2
 Merge button, 21-4
 properties, 21-4

- Properties button, 21-4
 - Remove button, 21-4
 - starting, 21-2
 - View button, 21-4
- Results Grapher Window, 21-3
- retrieving current attribute values, 8-13
- retry-interval for SNMP proxy, 19-6
- reverse order
 - graphing datasets, 21-7
- ribbon graphs, 21-7
- right-of-access, 11-4
- ring bell signal option, 15-19, 15-38
- rlogin, 14-23
- routers
 - adding names for multiple interfaces, 3-24
- RPC
 - definition of, Glossary-5
- RPC program number (for Grapher), 21-3
- RPC protocol, 1-3
- runtime database, 14-2, 18-2
 - clearing and initializing, 3-5
 - difference between .. and stored ASCII file, 3-38
 - guideline for size requirement for, 3-7
 - overview of, 3-1
 - saving, 3-37
 - saving to an ASCII file, 10-3
 - value of saving to ASCII file, 1-7
- runtime database files, 13-7
- runtime management database, 3-1

S

- Save menu
 - Management Database option, 14-9
 - Predefined Requests option, 14-10
- Save window, 17-26
- saving runtime database
 - value of, 1-7
- saving the runtime management database, 3-37
- schema
 - definition of, Glossary-5
- schema file
 - definition of, Glossary-5
- schema file for agents, 1-7
- schema file for element type, 1-6
- schema file locations, 17-25
- schema files, 13-6
 - agent, 18-2
 - element, 18-2
 - relationship among .. shipped with SNMP, 19-4
- schema list, 14-28
- schemas
 - SNMP, 19-3
- scrolling configuration, 17-6
- security, 11-1, 19-28
- security algorithm, 11-3
- security levels, 11-2, 11-4
- Select all
 - editing, 14-12
- selecting all elements, 14-12
- selecting requests, 15-7, 15-14, 15-33
- selecting settable agents
 - Set Tool, 24-4
- selecting settable groups
 - Set Tool, 24-4
- send
 - Data request, 14-22, 15-4
 - deferred reports, 15-47
 - Event request, 14-22
 - Predefined Data request, 14-22
 - Predefined Event request, 14-22
 - Quick Dump request, 14-21
- Send Mail signal option, 15-20, 15-39
- Send Mail to, 17-12
- Send to Program, 17-12
- Send to Program signal option, 15-20, 15-39
- sending deferred reports, 15-47
- sending mail
 - error report, 17-24

- sending reports
 - deferred, 15-6, 15-13, 15-26, 15-33
 - sending to program
 - error report, 17-25
 - sending trap report to program, 17-22
 - servers discover IPX/SPX, 23-1
 - services, discover IPX/SPX, 23-1
 - Set button
 - Set Tool, 24-4
 - Set Information list, 24-5
 - Set request
 - security, 11-1
 - setting attribute values, 14-22
 - set request, 8-18
 - Set Tool, 8-18
 - Agent field, 24-4
 - attribute list, 24-3
 - control panel, 24-3
 - control panel buttons, 24-4
 - control panel fields, 24-4
 - Get button, 24-4
 - Group field, 24-4
 - how to invoke, 24-1, 24-7
 - invoking from command line, 24-7
 - Key field, 24-4
 - Options field, 24-4
 - selecting settable agents, 24-4
 - selecting settable groups, 24-4
 - Set button, 24-4
 - Set Information list, 24-3, 24-5
 - Unset button, 24-4
 - window functions, 24-3
 - setting SNMP attributes, 8-18
 - Show Subview option, 14-25
 - showing graphs, 21-7
 - signal options
 - audio, 15-19, 15-38
 - blink glyph, 15-18, 15-37
 - dim glyph, 15-18, 15-37
 - mail options, 15-20, 15-39
 - play audio file, 15-19, 15-38
 - priority sets color, 15-19, 15-38
 - program, 15-20, 15-39
 - ring bell, 15-19, 15-38
 - Send Mail, 15-20, 15-39
 - Send to Program, 15-20, 15-39
- Simple Network Management Protocol (SNMP), 14-3
- SNMP hosts file
 - use for trap attribution, 19-19
- Snapshot utility, 4-31
- snm directory, 13-5
- SNM hosts view, 3-10
- SNM libraries, 13-6
- SNM log files
 - installation, 13-6
- snm+lock file, 13-7
- snm.conf file, 1-8, 8-2, 8-4, 8-8, 8-11, 8-12, 13-5, 13-6, 13-7, 19-4, 19-5, 19-6, 19-7, 19-10, 19-11, 19-13
- snm.glue file, 13-7, 18-2
- snm/bin directory, 13-9
- snm_cmd command, 13-8, 13-12
- snm_discover command, 22-2
- SNM_NAME environment variable, 13-7, 13-10
- snmdb+lock file, 13-7
- SNMDBDIR environment variable, 13-5, 13-7, 13-10
- SNMDEFAULTS environment variable, 17-30
- SNMDISCOVERMAP environment variable, 13-10
- SNMHOME environment variable, 13-5, 13-9
 - requirement for, 3-2
- SNMLINKMAP environment variable, 13-10
- SNMP, 14-1
 - community string, 24-4
 - definition of, Glossary-6
 - managing with, 8-1
 - setting trap priorities, 8-9
- SNMP attributes
 - setting, 8-18
- SNMP devices

- adding to database, 8-2
 - specifying color for, 22-27
- SNMP element
 - creating in database, 8-5
- SNMP get request, 8-13
- SNMP proxy
 - enterprise-specific traps, 19-14
 - receiving responses, 19-6
 - retry-interval, 19-6
 - trap handling, 19-12
- SNMP proxy agent, 1-2, 8-1, 19-1
- SNMP read community string, 8-6
- SNMP requests
 - redirecting to another proxy, 17-31
- SNMP schemas, 19-3
- SNMP set request, 8-18
- SNMP system description
 - mapping to element type, 22-28
- SNMP timeout, 8-6
- SNMP trap daemon, 19-12, 19-16
- SNMP trap file, 19-14, 19-16
 - glyph entries in, 19-18
- SNMP traps, 8-8, 19-12
 - interpreting, 19-23
- SNMP Vendor Proxy, 8-6
- SNMP view, 3-10
- SNMP write community string, 8-6
- snmp.hosts, 19-10
- snmp.hosts file, 13-8
 - when required, 19-6, 19-10
- snmp.schema, 19-3
- snmp.trapfile file, 13-8
- snmp.traps file, 8-12
- snmp-mibII.schema, 19-3
- snmp-mibII.schema file, 19-14
- SNMPv2
 - files, 19-29
 - man pages for, 19-27
 - SMI, 19-28
 - translation program, 19-29
- solid graphs, 21-7
- sorting requests, 15-44
- source code for selected agents, 13-6
- specifying events, 5-2
- specifying key, 24-4
- specifying options string, 24-4
- src directory, 13-6
- Start button
 - data report request, 15-11
- state changes of glyph, 5-27
- status of requests, 4-33
- stopped request state, 15-42
- stopping request state, 15-42
- stream
 - definition of, Glossary-6
- streams
 - copying to folder, 20-14
 - selecting, 20-11
- Streams menu, 20-10
- Strip Chart
 - data report, 16-12
 - data report request, 15-10, 15-29
- Strip Chart Properties window, 16-13
- Strip Charts, 4-15, 4-18
- struct directory, 13-6
- structure file
 - definition of, Glossary-6
- structure files, 18-2
- style of graphs, 21-7
- style of network layout, 12-20, 12-22
- subviews
 - changing, 14-25
- summary
 - Data and Event reports, 15-40
- Sun products with agents, 13-4
- Sun SNMP agent (snmpd), 2-10
- SunNet Manager
 - command-line options for, 3-5
 - requirement if installed in non-default location, 3-2
- supplied predefined data and event requests, 15-21
- switching views
 - Alarm Reports window, 6-4

symmetric layout style, 12-41 to ??
 granularity, 12-43
 incremental, 12-45
 node frame percent, 12-42
 start seed, 12-44
 threshold, 12-43
system
 definition of, Glossary-6

T

target host, 15-42
target system
 definition of, Glossary-6
Telnet, 14-23
three-dimensional graphs, 21-7
threshold
 event report request, 15-18, 15-37
threshold conditions
 event report request, 15-17, 15-36
threshold value
 event report request, 15-18, 15-37
time stamps
 graphing, 21-6
timeout, 8-6
timeout for requests, 17-7
title of Console windows, 17-5
Tool Properties option, 20-15
Tools
 Glyph menu, 14-23
 rlogin to target system, 14-23
 Telnet, 14-23
Tools button, 14-18
Tools menu
 Discover, 22-2
 Grapher, 21-2
tools menu
 adding submenus, 10-10
 modifying, 10-10
Tools menu - definition, 18-2, 18-13
Tools menu for element type, 10-12
trap audio signal, 17-21
trap effect, 17-21

trap effect propagation, 17-21
trap file, 19-14
 glyph entries, 19-18
trap glyph changes, 5-27
trap report
 mailing, 17-22
 sending to program, 17-22
trap reports, 6-15
 maximum number, 17-28
trap type
 authenticationFailure, 19-12
 coldStart, 19-12
 egpNeighborLoss, 19-12
 enterprise-specific, 19-12
 linkDown, 19-12
 linkUp, 19-12
 warmStart, 19-12
Traps
 when database file loaded, 17-32
traps, 13-8, 19-12
 attributing to pseudo-devices, 19-17
 changes to database, 17-22
 forwarding, 8-11
 interpreting, 19-23
 receiving, 8-8
 setting filters for, 8-10, 19-7
 setting SNMP trap priorities, 8-9
traps, receive from SNM database, 12-12
troubleshooting, 12-10, 12-13
two-dimensional graphs, 21-7
type of element, 1-6
type of request, 15-41

U

Undefine button
 data report request, 15-30, 15-40
undo button, 12-8
Unset button
 Set Tool, 24-4
user command definition, 18-8
user commands, 18-2
User Commands menu, 18-8

username display, 17-5

V

verification of Console quit, 17-27

verify function
definition of, Glossary-6

Vertical Scrolling, 17-6

view

Data and Event reports, 16-9, 16-19

definition of, Glossary-6

Event/Trap reports, 16-9, 16-19

glyph state, 17-18

view displayed, 17-17

view element type definition, 18-6

view hierarchy, 3-16

View menu, 4-15

Add Background, 16-18

Alarms Reports, 16-2

Clipboard, 16-19

Data Reports, 16-9

Error Reports, 16-16

Event/Trap Reports, 16-14, 19-22,
19-25

Events, 16-17

Find, 16-18

Remove Background, 16-18

view name in membership record, 18-10

viewBackground instance record, 18-12
for background image, 10-5

viewing data, 4-15

viewing data in Graph Tool, 4-17

viewing data in Indicators, 4-18

viewing data in Strip Charts, 4-18

viewing Data Reports window, 4-16

viewing error messages, 6-11

viewing requests, 4-33

viewing trap reports, 6-15

views, 18-3, 18-10

adding background images, 10-1

background images, 18-12

changing, 14-25

glyph, 18-3

glyph state propagation, 5-30

high-level, of network, 12-10

moving elements between, 3-32

navigating, 3-16, 14-19

of networks, 12-1

print views of network, 12-14

SNM hosts, 3-10

SNMP agents, 3-10

SNMP devices, 3-10

using monitor function for non-
standard .., 22-29

views, switching, 6-4

visual signal options

blink glyph, 15-18, 15-37

dim glyph, 15-18, 15-37

event report request, 15-18, 15-37

priority sets color, 15-19, 15-38

W

Window Title, 17-5

write community, 8-6

write-security, 11-1

X

X resource configuration, 17-30

X terminal, 14-2

X11 protocol, 14-2

X11 server, 14-2

Z

zooming graphs, 21-10

Copyright 1996 Sun Microsystems Inc., 2550 Garcia Avenue, Mountain View, Californie 94043-1100, U.S.A. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou de sa documentation associée ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a.

Des parties de ce produit pourront être dérivées du système UNIX[®] licencié par Novell, Inc. et du système Berkeley 4.3 BSD licencié par l'Université de Californie. UNIX est une marque enregistrée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company Ltd. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Sun, Sun Microsystems, le logo Sun, Solstice, Solstice Site Manager, Solstice SunNet Manager, Solstice Domain Manager, and Cooperative Consoles sont des marques déposées ou enregistrées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC, utilisées sous licence, sont des marques déposées ou enregistrées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

Les interfaces d'utilisation graphique OPEN LOOK[®] et Sun[™] ont été développées par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant aussi les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

Le système X Window est un produit du X Consortium, Inc.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" SANS GARANTIE D'AUCUNE SORTE, NI EXPRESSE NI IMPLICITE, Y COMPRIS, ET SANS QUE CETTE LISTE NE SOIT LIMITATIVE, DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DES PRODUITS A RÉPONDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ILS NE SOIENT PAS CONTREFAISANTS DE PRODUITS DE TIERS.