

InterMapper 3.6: Server and Network Monitoring

InterMapper is a network monitoring and alerting program. It continually tests routers, servers, hubs, and other computer devices that are attached to your network. If InterMapper detects a failure, it will send notifications to one or more individuals via sounds, e-mail, pagers, or by running a program to correct the problem.

This manual is divided into sections: To learn about how to get InterMapper up and running quickly, read the Tutorial chapter. Read the other chapters to get detailed information about specific features of the program.

[Chapter 1 -- InterMapper Tutorial](#)

[Chapter 2 -- Command and Menu Reference](#)

[Chapter 3 -- Notifications](#)

[Chapter 4 -- Preferences](#)

[Chapter 5 -- Information Windows](#)

[Chapter 6 -- Web Server](#)

[Appendix A -- Built-in Probe Reference](#)

[Appendix B -- Customizing InterMapper's Probes](#)

[Appendix C -- Customizing Web Pages](#)

[Appendix D -- Files and Folders](#)

[Appendix E -- Frequently Asked Questions](#)

You can read this manual a page at a time by clicking the [Next](#) | [Previous](#) links. They will take you to the next (or previous) page in the outline shown at the left.

Please give us comments at the address listed below. Thanks!

Rich, Bill, Stuart, Tex, and John

[Dartware, LLC](#)

[InterMapper Feedback](#)

Preface

A Brief History of InterMapper

InterMapper is a network monitoring and alerting tool. It was initially developed at Dartmouth College where Bill Fisher and Rich Brown worked to create a tool that would monitor the College's locally-developed New England Digital (NED) AppleTalk and IP routers. These minicomputer-based routers had extremely limited memory, and thus couldn't ever be programmed to speak SNMP. With more than one hundred of these routers in the basements of buildings on campus, the College decided to write its own tool for monitoring the network. As more SNMP-speaking commercial equipment was brought on campus, InterMapper was extended to support SNMP, and later other probe types.

The program was good enough that Rich and Bill were encouraged to market InterMapper commercially beginning in July 1996. (They had some practice marketing software from their experience selling the MacPing software from 1992.) Dartmouth also began selling their SNMP Watcher MIB console in March 1999.

In April, 2000, Dartmouth College transferred title to InterMapper, MacPing, and SNMP Watcher to a newly-formed company, Dartware, LLC. The founders were Rich Brown, Bill Fisher, and Stuart Pompian, an area businessman. Dartmouth College retains a share of the ownership of Dartware which will continue development and marketing of those software products. New to the InterMapper team are Tex Clayton, programmer, and John Sutton, Customer Service.

In July 2000, Dartware released InterMapper 3.0. In July 2001, they introduced version 3.5. January 2002 introduced InterMapper 3.6, which was the first version to support InterMapper Remote.

Credits

Author: Bill Fisher

Product Manager: Rich Brown

Inspiration: Stuart Pompian

We have had the good fortune to receive suggestions, bug reports, etc from Nick Alex, Maria Arista, Rich Battin, Mel Beckman, J rgen Br ndle, Steve Campbell, Frank Charbonnier, Charlie Clark, Mike Cullum, Joe Drees, Jeff Donovan, Greg Dunham, Steve Erde, Jeffrey Flaherty, Alain Fontaine, Evan Gamblin, Ron Gee, Doug Grinbergs, Gulfie, Brian Hall, Brandon Handeland, Stewart Harris, Paul Hill, Dale Hofstetter, Valentin Horveilleur, Joakim Jardenberg, Jeff Kell, Shahid Khan, Austin Kinsella, Mike Lieberman, Jim Madden, Matthew Marino, Mark Maytum, Christopher McCrory, Bill McHargue, Kevin Miller, Craig Oshiro, Dennis O'Reilly, Chris Pepper, Mark Persiko, Jakob Peterh nsel, Dave Reddy, Mike Richardson, Laura Schlegel, Willy Schley, Tim Streater, John Simmonds, Jerry Segers, Dan Sokol, Matt Stevens, Pat Storr, Alan Sutter, Andrew Trombettas, Al Tufts, Tim Urban, Bennie Warren, Doug Weathers, Mark Weisler, and Nick Wesselman

Version 3.0 benefitted greatly from the following people's ideas, thoughts, and bug reports: Steve Erde, Chris Pepper, Tim Streater, Steve Campbell, Charles Clark, Maria Arista, Sean Dunten, Geoff Bronner, Peter Thompson, Dennis O'Reilly, and Al Tufts.

Thanks to the following people for their suggestions, bug reports, code snippets, cheer leading, and patience: Maria Arista, Steve Campbell, Charles Clark, Stan Dunten, Zyg Furmaniuk, David Gelhar, Bill Huschle, Jim Matthews, Kee Nethery, Chris Pepper, Frostie Sprout, Tim Streater, Punch Taylor, Pat Wilson, and Bill Woodcock.

As always, many, many thanks to our alpha testers and beta testers!

Trademark and Copyright Information

InterMapper is a registered trademark of Dartware, LLC. InterMapper Remote, MacPing, and SNMP Watcher are trademarks of Dartware, LLC. InterMapper, InterMapper Remote, MacPing, and SNMP Watcher are copyright © 1996-2002 by Dartware, LLC. All rights reserved.

Other programs mentioned in this manual are trademarked and copyrighted by their respective owners.

This manual documents InterMapper 3.6, and was last updated on Friday, January 25, 2002.

Dartware, LLC
25 South Main Street
PO Box 130
Hanover, NH 03755-0130 USA
InterMapper@dartware.com
<http://www.dartware.com>

What's New in InterMapper 3.6?

InterMapper 3.6 contains a number of new features. These include:

- **InterMapper Remote (Kali) support.** InterMapper Remote is a remote viewer application that allows you to see live copies of your maps on a machine across the internet. [details...](#)
- **Background images for each map.** Choose Map Settings from the Edit menu. [details...](#)
- **Response time measurements.** Probes now measure response time and display the measurement in popup windows. TCP probes also allow timing of responses within the script. [details...](#)
- **New strip chart options.** You can now add an unlimited number of values to a strip chart to be logged and saved to a text log file. The displayed colors, line styles, and legends can now be customized. [details...](#)
- **Persistent Chart Data.** InterMapper now writes chart data to disk, and restores it after quitting and re-launching. This allows unlimited amounts of strip chart data. [details...](#)
- **Outages window.** The new Outages window lists the down & up time of all outages that InterMapper has detected. [details...](#)
- **Separate Log Files.** InterMapper can write different kinds of log information to separate log files. This allows the network manager to separate messages about major failures (UP & DOWN messages, for example) from Web and telnet access log entries, etc. [details...](#)
- **SNMPv2c and 64-bit Counters:** InterMapper now polls SNMPv2c MIB variables for devices that support 64-bit counters. This allows an accurate display of the speed of very high speed interfaces. [details...](#)
- **New Probe Facilities:** There is a new probe that tests Simple Network Pager Protocol (SNPP) servers. In addition, Custom SNMP probes can issue Get-Requests (instead of Get-Next-Requests), and disable InterMapper's normal MIB-II queries for devices that don't implement that MIB. [details...](#)
- **Faster Poll Rates:** If you wish to poll devices faster than the default 30 seconds, you can now select 1, 5, and 10 second intervals from the Poll Interval popup menu. [details...](#)
- **Alternate SMTP Host:** You can now specify two SMTP hosts so that InterMapper can still send its e-mail notifications, even if the primary host is down. [details...](#)

Our lawyers made us say this...

Software License Agreement

This is a legal agreement between you and Dartware, LLC covering your use of InterMapper ,InterMapper Remote (tm), SNMP Watcher (tm), MacPing (tm), and other Dartware products and the associated documentation (the "Software"). Be sure to read the following agreement before using the Software. BY USING THE SOFTWARE (REGARDLESS IF YOU HAVE REGISTERED THE SOFTWARE OR NOT), YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT USE THE SOFTWARE AND DESTROY ALL COPIES IN YOUR POSSESSION.

The Software is owned by Dartware, LLC and is protected by United States copyright laws and international treaty provisions. It is licensed to you. Therefore, you must treat the Software like any other copyrighted material (e.g., a book or musical recording).

License

This license allows you the right to use one copy of the Software on a single computer. You may make one (1) copy of the software as a backup. You may not network the Software or otherwise use it or make it available for use on more than one computer at the same time. You may not rent or lease the Software, nor may you modify, adapt, translate, decompile, or disassemble the Software. If you violate any part of this agreement, your right to use this Software terminates automatically and you must then destroy all copies of the Software in your possession.

Disclaimer of Warranty

The Software and its related documentation are provided "AS IS" and without warranty of any kind. The person using the software bears all risk as to the quality and performance of the software. If you paid for the product, and within 90 days find that it doesn't do what you want, then you can notify Dartware, LLC and your money will be refunded and your license canceled. Dartware, LLC hereby disclaims all warranties relating to this software, whether express or implied, including without limitation any implied warranties of merchantability or fitness for a particular purpose. Dartware, LLC will not be liable for any special, incidental, consequential, indirect or similar damages due to loss of data or any other reason, even if Dartware, LLC or their agent has been advised of the possibility of such damages. In no event shall Dartware, LLC be liable for any damages, regardless of the form of the claim.

US Government

Government End Users: If you are acquiring the Software on behalf of any unit or agency of the United States Government, the following provisions apply. The Government agrees: (i) if the Software is supplied to the Department of Defense (DoD), the Software is classified as "Commercial Computer Software" and the Government is acquiring only "restricted rights" in the Software and its documentation as that term is defined in Clause 252.227-7013(c)(1) of the DFARS; and (ii) if the Software are supplied to any unit or agency of the United States Government other than DoD, the Government's rights in the Software and its documentation will be as defined in Clause 52.227-19(c)(2) of the FAR or, in the case of NASA, in Clause 18-52.227-86(d) of the NASA Supplement to the FAR. The manufacturer is Dartware, LLC, 25 South Main Street, PO Box 130, Hanover, NH 03755-0130 USA.

This Agreement shall be governed by the laws of the State of New Hampshire. If for any reason a court of competent jurisdiction finds any provision of the Agreement, or portion thereof, to be unenforceable, that provision of the Agreement shall be enforced to the maximum extent permissible so as to effect the intent of the parties, and the remainder of this Agreement shall continue in full force and effect.

Acknowledgement

You acknowledge that you have read this agreement, understand it, and agree to be bound by its terms and conditions. You further agree that it is the complete and exclusive statement of the agreement between you and Dartware, LLC which supercedes all proposals or prior agreements, oral or written, and all other

communications between you and Dartware, LLC relating to the subject matter of this agreement.

Dartware, LLC

25 South Main Street, PO Box 130

Hanover, NH 03755-0130 USA

Voice: 603-643-2268; Fax: 603-643-2289

Web: <http://www.dartware.com>

InterMapper Tutorial

This Tutorial chapter shows how to install InterMapper and start monitoring your network.

The easiest way to start out is to launch InterMapper and follow along with this tutorial. Each of the pages listed below covers a separate topic. In 15-20 minutes, you'll have a functional map of your network!

[Installing and Launching InterMapper](#)

InterMapper is easy to install. There's an installer for the Classic MacOS, and a disk image for MacOS X.

[Starting a Map](#)

InterMapper automatically discovers the routers and subnets on a network. It can then scan those subnets to find the servers, hosts, workstations, etc. connected to the network.

[Adding Devices](#)

If you have a list of device names or addresses in a separate file, it's easy to paste them into a map.

[Arranging the Map](#)

Once there are devices on the map, you can move them around to suit your needs.

[Monitoring the Network](#)

InterMapper automatically monitors everything in the network, even while you're arranging the map. This section describes how InterMapper displays various problems in the network.

[Pop-up Windows](#)

Clicking and holding on a device, link, or network pops up a window that gives detailed information about that item. You can also tear off the window to keep it open permanently.

[Strip Charts](#)

InterMapper creates strip charts that give a historical view of various statistics in the network. Each map can have multiple strip charts.

[Modifying Device and Network Labels](#)

By default, InterMapper labels an item with its DNS name or IP address. This section describes how you can customize the labels of items on the map.

[Adding Switches to a Map](#)

InterMapper provides detailed information about switches. This shows how to add a switch and get the most information from it.

[Adding Additional Network Ovals](#)

InterMapper has a facility to add additional IP subnets or AppleTalk networks to a map. Also includes a subnet mask table.

[Scanning for Additional Devices](#)

InterMapper can automatically scan a network's address range for devices. It then connects the device(s) to the proper place in the map.

[Using Helper Programs](#)

With the appropriate helper programs installed, InterMapper can send pings, perform a trace route, sending SNMP queries, connect with web and telnet to other devices, or use Timbuktu to administer a remote computer.

Installing and Launching InterMapper

InterMapper is a Macintosh application that creates maps of your IP and AppleTalk networks and then monitors everything - the servers, routers, network equipment, workstations and other devices - on those networks. *InterMapper* will report when devices fail and when they recover. It will also show traffic levels, error statistics, and utilization of links.

If the devices in your network speak the Simple Network Management Protocol (SNMP), *InterMapper* can automatically determine the interconnections within your network and create the map by itself. It also scans subnets to discover and add devices to the map. *InterMapper* also has flexible drawing tools that enable you to rearrange its map to suit your needs.

It's easy to set up an *InterMapper* map. We recommend that you read through this tutorial: in fifteen to twenty minutes, you'll have a functioning network monitor, and you'll have discovered the major features of *InterMapper*. So let's begin...

Installing and Launching InterMapper

InterMapper is available for both the Classic MacOS and MacOS X. Use the appropriate installation procedure for your version.

Classic *InterMapper* requires System 7.5 or newer, as well as Open Transport 1.1.2 or newer, properly configured for your network. Any Macintosh computer with 16 or more megabytes of memory will run *InterMapper*.

InterMapper for MacOS X requires a computer running MacOS 10.0 or newer. No additional hardware or software is necessary.

Classic MacOS (7.0 to 9.2)

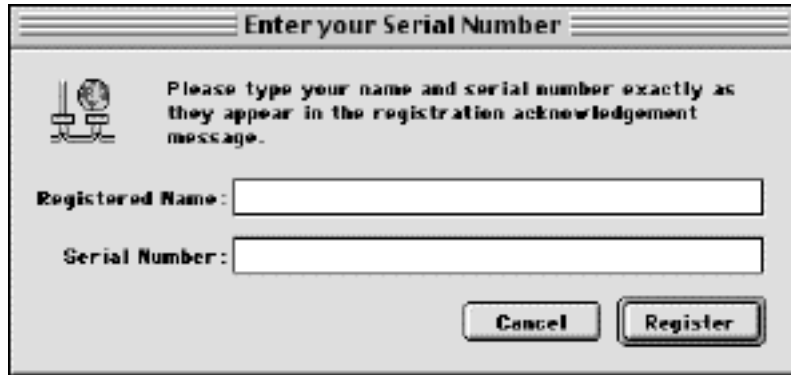
1. Double-click the Installer program. You may install InterMapper in any folder on any disk. It does not require or install any extensions or control panels.
2. You will be asked whether you want InterMapper to launch whenever the computer starts up. Click **Yes** to place an alias of the program in the *Startup Items* folder. If you click **Cancel** no alias will be installed.
3. The InterMapper 3.5 folder will open. If you are upgrading from a previous version of InterMapper, drag *copies* of your map files into this new folder. (To make copies of the files, hold down the Option key as you drag them to the new location.)
4. Double-click the *InterMapper* application to launch it.

InterMapper for MacOS X

1. The downloaded version of InterMapper is a disk image file. Double-click the .dmg file: it will appear as a disk on your desktop.
2. Open the disk on your desktop and drag the *InterMapper* folder to any folder on your computer. (We recommend the Documents folder in your home directory.)
3. If you are upgrading from a previous version of InterMapper, drag *copies* of your map files into this new folder. (To make copies of the files, hold down the Option key as you drag them to the new location.)
4. Double-click the *InterMapper* application to launch it.
5. InterMapper must be given permission to access reserved network resources. Click the **Grant Network Privileges...** button. You will be prompted to enter the name and password of an Administrator. (This is a one-time operation.)
6. You may also want InterMapper to start up when you log in. To do this, click the Login icon of the System Preferences. Drag InterMapper's icon to the window. You can also check the **Automatically log in** box so that InterMapper will begin running after a reboot.

The downloadable version of InterMapper will run for only one hour. You'll need a serial number to use for longer.

If you already have a serial number (because you purchased the software or you went to the evaluation form), click the "Enter your serial number now" link in the yellow window shown at the right. You can then type your name and serial number into the Registration Window shown below.



About Serial Numbers

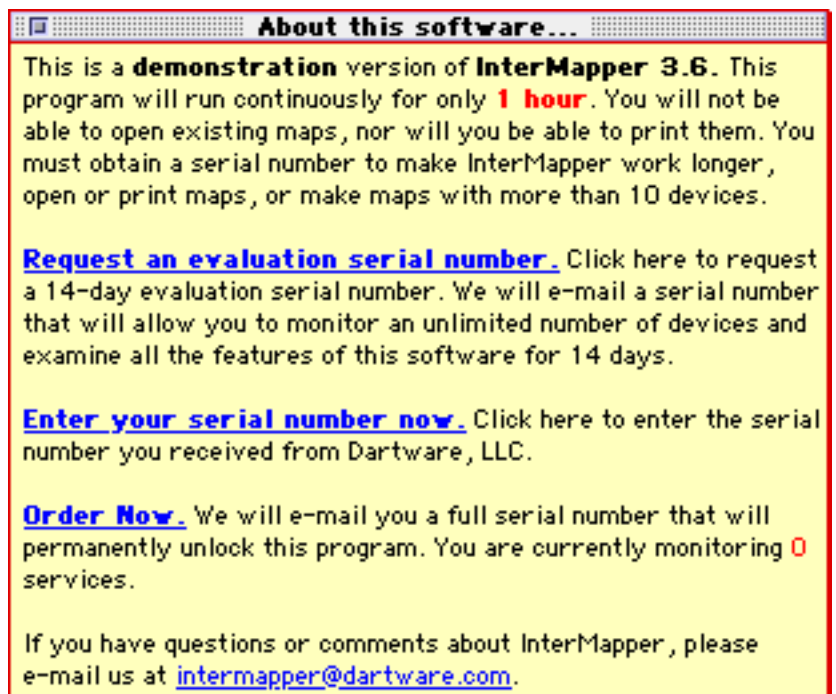
The downloadable version of InterMapper is a demonstration sampler. It will operate only for one hour, tests only 10 devices, and cannot open saved maps.

To evaluate InterMapper for a longer period, you must get a serial number. The easiest way to do this is to click on the "Request an Evaluation Serial Number" link in the yellow window shown at the right. It will open a web form on your web browser. We will e-mail you a serial number that allows you to use InterMapper to monitor an unlimited number of devices for 14 days. Use the Registration window (described above) to enter the your name and the serial number.

When you purchase InterMapper, we will send you a full serial number that unlocks the software permanently.

If the demo or evaluation software times out, InterMapper simply ceases to operate. It will *never* delete or alter files on your hard drive.

Now just follow along in this Tutorial to try out InterMapper's features.



About this software...

This is a **demonstration** version of **InterMapper 3.6**. This program will run continuously for only **1 hour**. You will not be able to open existing maps, nor will you be able to print them. You must obtain a serial number to make InterMapper work longer, open or print maps, or make maps with more than 10 devices.

[Request an evaluation serial number](#). Click here to request a 14-day evaluation serial number. We will e-mail a serial number that will allow you to monitor an unlimited number of devices and examine all the features of this software for 14 days.

[Enter your serial number now](#). Click here to enter the serial number you received from Dartware, LLC.

[Order Now](#). We will e-mail you a full serial number that will permanently unlock this program. You are currently monitoring 0 services.

If you have questions or comments about InterMapper, please e-mail us at intermapper@dartware.com.

Starting Your Map

(Letting InterMapper Autodiscover the Network)

When it launches, *InterMapper* will open an empty map window, ready to discover the topology of the network. You'll need to enter a *starting point* - the DNS name or IP address of a host or router where autodiscovery should begin. You may use the drawing tools to arrange the map to your liking. Finally, you can place *InterMapper* in its monitoring mode so that it will alert you to problems.

InterMapper uses a combination of heuristic techniques, including SNMP probes, ICMP and AppleTalk echo packets, and DNS and NBP queries, to discover devices that are present, and then places those devices on a map.

The window shown in Figure 1-1 controls autodiscovery. Enter a host name or IP address as the starting point for autodiscovery. InterMapper suggests the DNS name (or IP address) of the router, or the computer it's running on if there's no router. You may enter any DNS name or IP address if you want to create a map of another part of a network. If you enter the name or address of an SNMP-speaking router, InterMapper will draw interconnections to other routers in the network more quickly.

You control the starting point, the community string, the breadth of the network search, and the kinds of devices that will be automatically added to the map using this window. The options are:



Figure 1-1: The Autodiscovery dialog

Starting host name: This is the name or address of a device that should begin the autodiscovery.

Specify a SNMP community string: allows you to specify an additional SNMP Read-only community string to be used to interrogate all devices. (InterMapper always attempts to read SNMP information using the default 'public' community string.)

Stay within __ hops of starting device stops autodiscovery after InterMapper has searched the specified number of hops from the starting device.

Scan for devices on all networks: allows you to specify which kinds of devices should be automatically added to the map. Checking this box, or clicking the **Filter...** button opens the Network Scanning window shown in Figure 1-2.

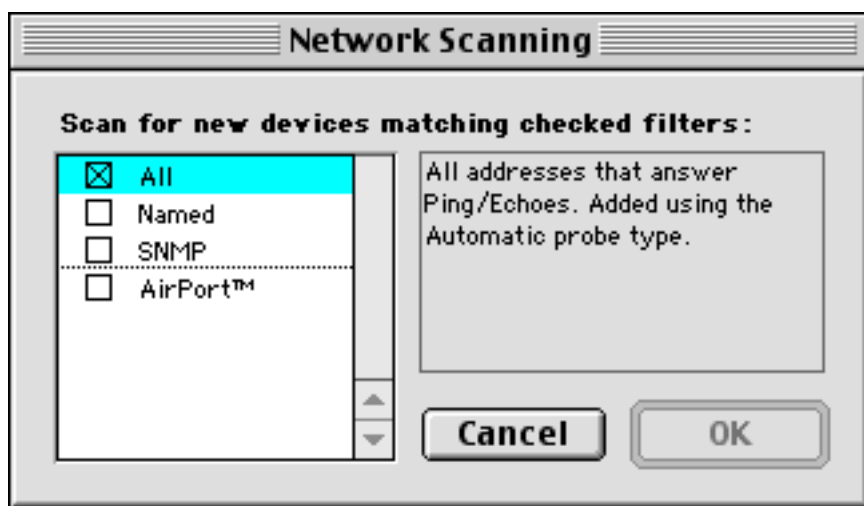


Figure 1-2: The Network Filter dialog. Check the boxes to add the associated type of device to the map.

These options are:

- **All** forces a complete IP address scan for each network
- **Named** Each IP address is looked up in the DNS. If a corresponding name is present, the device is added to the map
- **SNMP** InterMapper sends a SNMP GetRequest to each address in the range. Devices that respond are added to the map.
- **AirPort** If the SNMP response indicates it's an Apple AirPort Base Station, the device is added to the map.

During autodiscovery, InterMapper attempts to discover the subnet of the starting device. It then performs the following processes concurrently and iteratively until the specified limits are reached:

- InterMapper checks each discovered device to see if it is an SNMP-speaking router. If so, InterMapper attempts to discover what interfaces it has, and what other routers are connected to those interfaces. InterMapper will iteratively query the discovered routers for their connected networks, and then begin performing autodiscovery on those networks.
- For each network or subnet discovered, InterMapper pings every address on that subnet to find more active or named devices.
- InterMapper uses several heuristics to characterize each discovered device. For example, it sends SNMP queries (with the 'public' and Additional SNMP community string specified in the Autodiscovery window) to determine what kind of device is present.

Warning: In autodiscovery mode, InterMapper may ping or query every device address on a subnet. This may set off certain intrusion detection alarms that may be installed on the network. Be sure to check with the network manager before using this feature.

To provide visual feedback about the autodiscovery progress, *InterMapper* displays the floating window shown in Figure 1-3. In addition, it changes the color of the networks being autodiscovered to purple. When *InterMapper* has found all the devices within its bounds, it changes the button to read "Done". You may click the "Stop" button to halt the autodiscovery. (InterMapper will continue to characterize and add to the map any devices it has discovered, but it will not add any new devices or networks after you click Stop.)



Figure 1-3: The progress windows during autodiscovery.

Figure 1-4 shows a typical map after autodiscovery has completed.

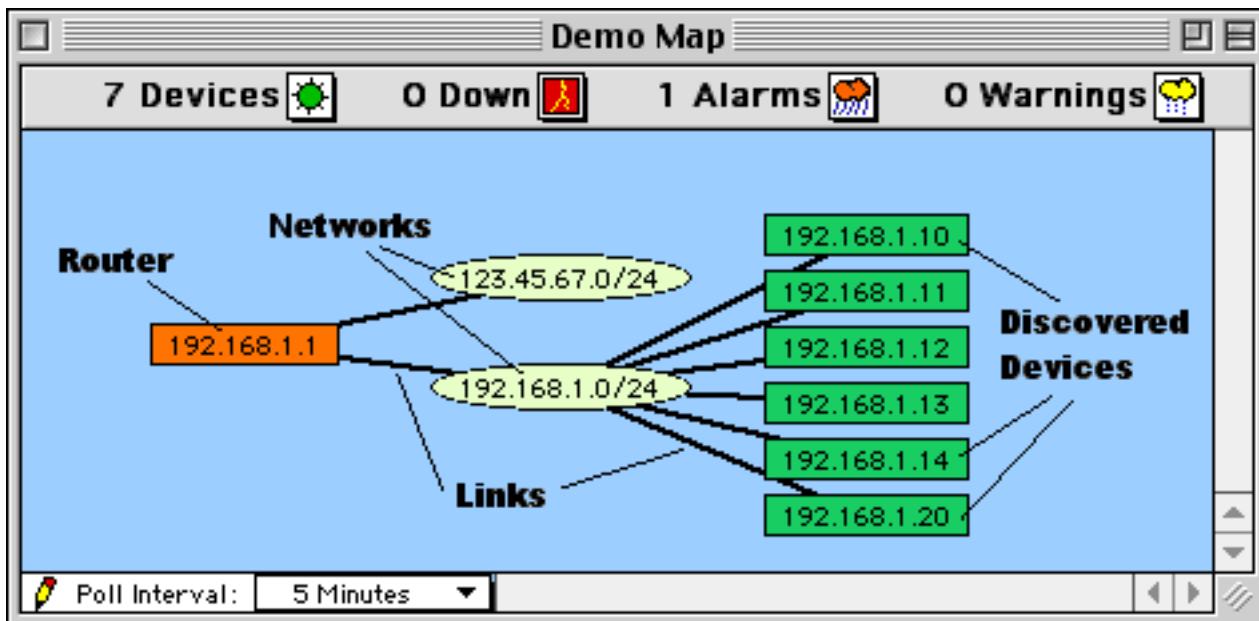


Figure 1-4: Autodiscovered devices and networks. Routers are interconnected by links to networks. Devices that have been autodiscovered are shown at the right. The top of the window whows the number of devices in the map, and a summary of their status.

InterMapper displays devices with their full DNS or AppleTalk names by default. Networks are shown with both an IP address (and the number of bits in the subnet mask) and an AppleTalk network number (or network range) if applicable. For example, the network labeled `192.168.1.0/24 142` in Figure 1-4 indicates that the IP devices are in subnet `192.168.1.0`, with a subnet mask of 24 bits (`255.255.255.0`). The AppleTalk network number on that same segment is 142.

Adding Other Devices

You may add devices manually using the **Add Device(s)...** command from the **Network** menu. This opens the **Add Device(s)...** window shown in Figure 1-5 to allow you to enter the device names and/or addresses.



Figure 1-5: Add Device(s) window. Type or paste a list of DNS or AppleTalk NBP names or IP or AppleTalk addresses into this window. After you click OK, the items will be added to the map.

Enter the device names/addresses in the field as shown. You may type individual host names or addresses, or paste the host names into this field. Entries must be separated by commas or by whitespace characters (spaces, tabs, or carriage returns). The list of hosts might come from a text file, from the *MacPing* program, or from a traceroute program.

The **Probe Type:** popup menu allows you to select what kind of probe will be used with the device. If the probe type requires a port number, its default will be displayed in the **Port:** field; you may change this as needed.

You may optionally enter a SNMP community string to be used as an alternative to 'public'.

Click **OK** to add the devices to the map.

If any of the device names cannot be resolved (e.g., if they are not configured in your domain name system server) or if a device cannot be tested with the selected probe, you will be given a chance to correct the entry.

Click the question mark icon in the lower left corner to get help. The information will appear in your web browser's window.

Arranging the Map

Once you've added all the devices to the map, you can arrange them to suit your ideas about the network.

By default, *InterMapper* displays *devices* as rectangles in its map. These devices are connected by *links* - straight lines of differing thickness to indicate the kind of link - to *networks*, which are represented as ovals.

Use *InterMapper's* layout tools to arrange the map to your liking. The strategy we recommend is to find one or more *clusters* of related items and move them close together. Once you have created clusters, you can move them to different parts of the map. For example, an Ethernet or FDDI backbone with its attached routers might make a good cluster. Similarly, a central router or switch with its attached networks might serve as a cluster.

To see the interconnections between devices more easily, use the **Cycle** tool. This spreads out the items in the map, and makes the relationships more clear. To do this, choose **Select All** (Cmd-A) from the **Edit** menu, then choose **Cycle** from the **Layout** menu.

The **Cycle** command moves all devices and networks near the edge of the window as shown in Figure 1-6. You may want to make the window a bit larger to leave more room for moving items around.

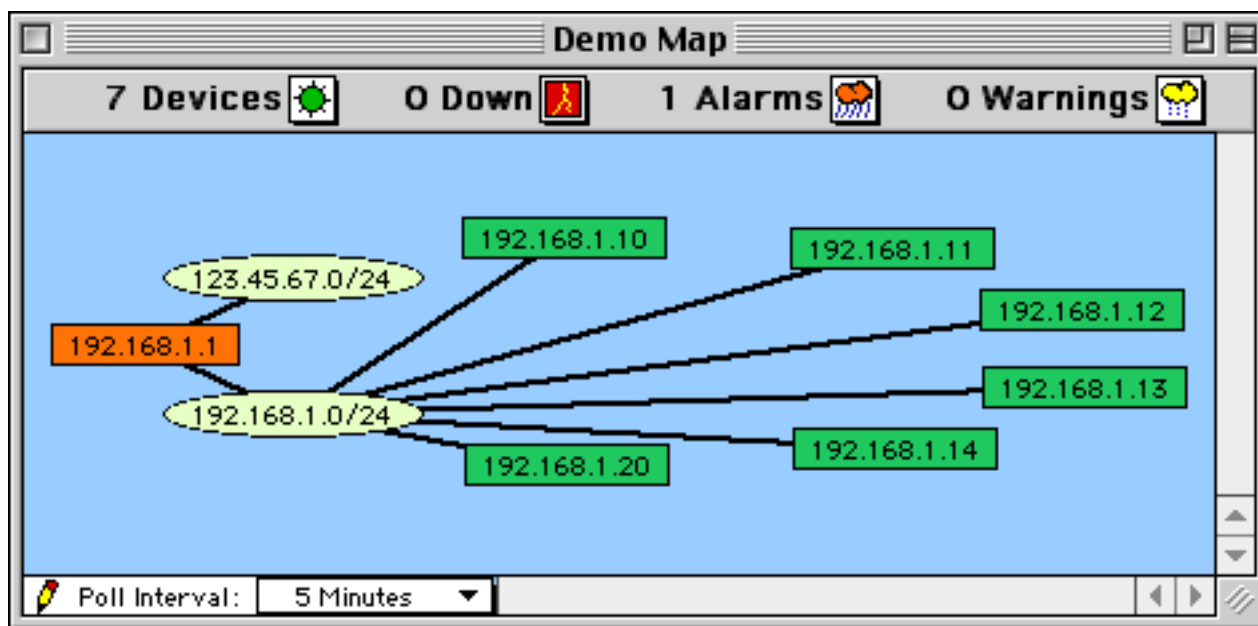


Figure 1-6: Results of a Cycle layout.

Now look for items with lots of connections. For example, the network (oval) labeled "192.168.1.0/24" in the lower-left corner has many connections. To see these connections better, drag the oval around the map: all the lines remain attached (like rubber-bands). Figure 1-7 shows that the network has been moved closer to the center of the window.

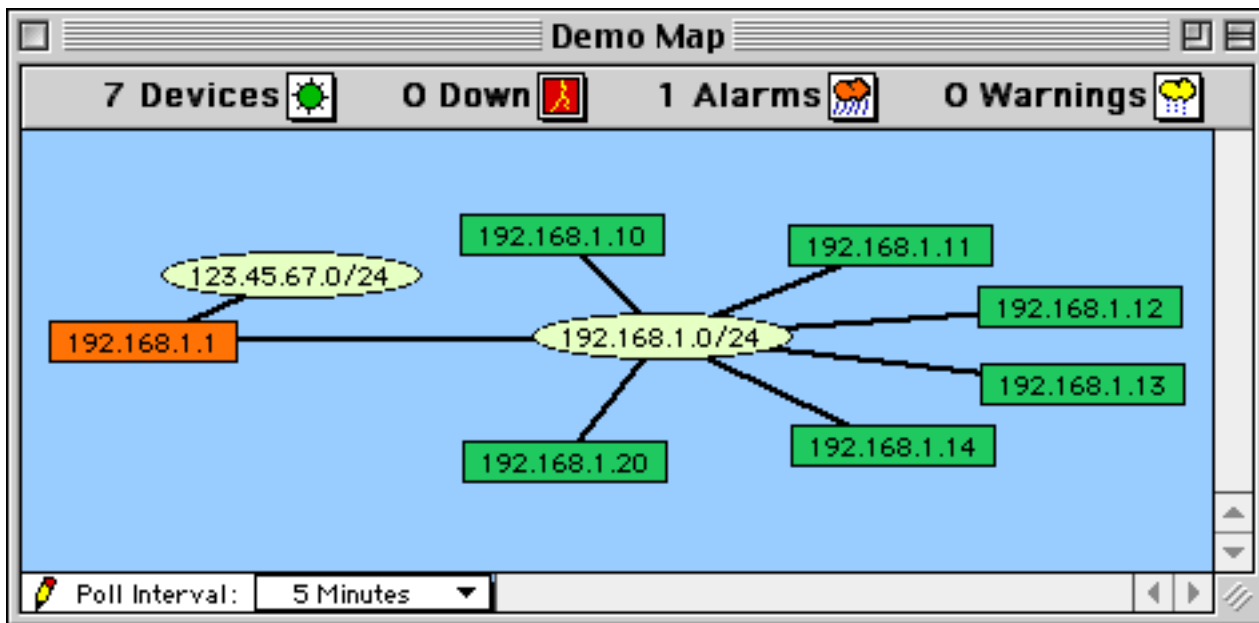


Figure 1-7: The network (oval) labeled 192.168.1.0/24 has been dragged from the lower left to the center of the window to show its interconnections.

This particular oval happens to represent an Ethernet segment that interconnects several devices in an office. To make it a cluster, you can use the **Bus** command from the **Layout** menu.

Select the oval identified above, and choose **Bus**. The oval changes to a vertical line and all its attached items are aligned vertically, as shown in Figure 1-8.

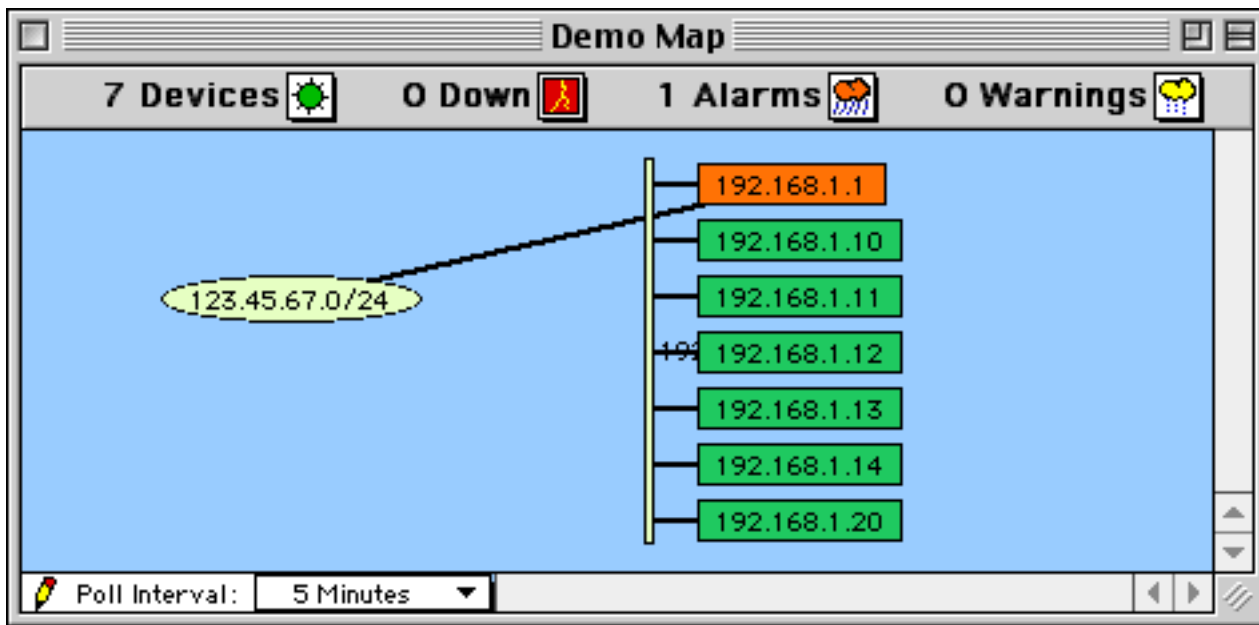


Figure 1-8: Items connected by a Bus network. The vertical line is the network 192.168.1.0/24, which was previously displayed as an oval.

Alternatively, you may choose **Star** from the **Layout** menu. The connected items will be arranged in a circle. In this example, the 192.168.1.0/24 network is the center surrounded by a star of the devices attached to it. Figure 1-9 shows the effect of the Star command:

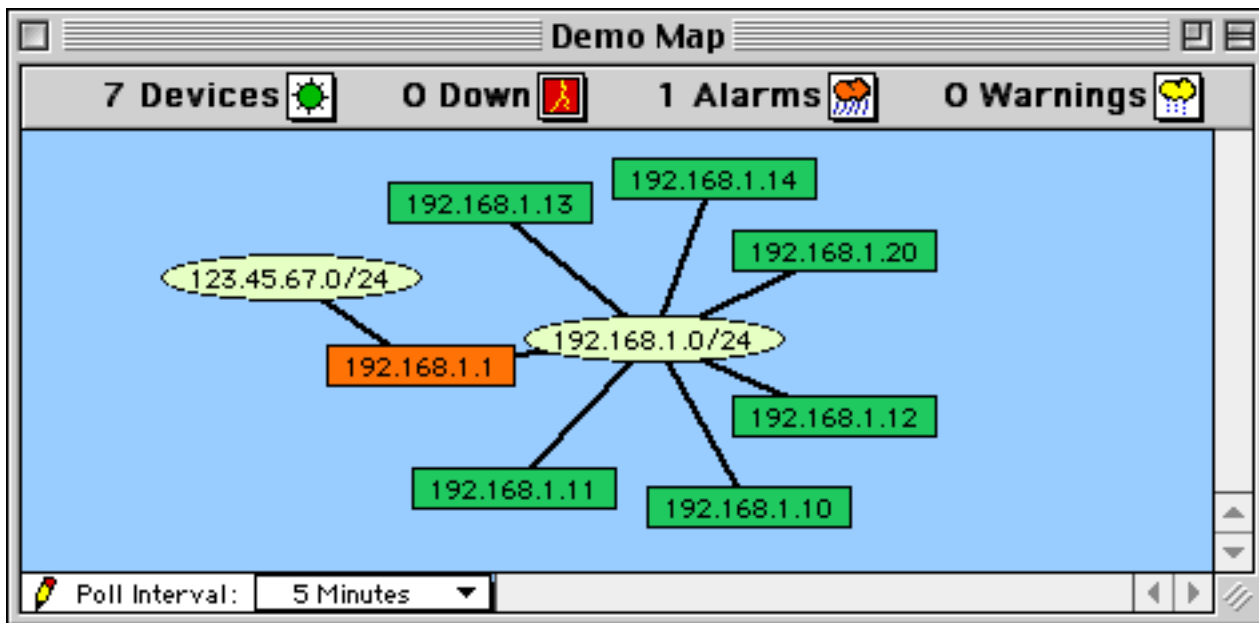


Figure 1-9: The Star command moves items in a circle surrounding the router.

Having formed a bus or star cluster, drag it to the edge of the window. This allows you to see the interconnections of the remaining devices. Create other clusters as required.

Once you have identified and arranged the clusters, use the following tips to fine-tune your map:

- You may move one or more items around the window by dragging them to a new position. Use shift-click to add or remove items from the current selection before dragging.
- *InterMapper* provides an automatic method of selecting adjacent items. Option-click on an item to select all the items connected to it. If you option-click again, the next-most adjacent items will be added to the selection.
- The **Layout** menu commands affect *placement* of items in the map. In addition to the **Cycle**, **Bus**, and **Star** commands described above, you can use the following menu commands to alter the orientations or sizes of the items in the map.
 - Align...** to modify the alignment of items
 - Rotate...** to rotate the selected items around their center
 - Scale...** to increase or decrease the separation of the selected items
- The **Display** menu commands affect the *appearance* of individual items. These commands include:
 - Shape:** change the item's shape to a rectangle, oval, wire, cloud, text, or other icon
 - Color:** pick a color for an item in the map
 - Label:** modify a text label for an item in the map
 - Font:** change the font of a text label
 - Size:** change the size of a text label
 - Style:** change the style of a text label
 - Label position:** change the location of a text label relative to its item
- If networks or ports are not important for a map, hide them with the **Hide Selection** command in the **Edit** menu. If you simply delete them, they will be reappear when the map is next opened.
- See the [Modifying Device and Network Labels](#) and [Adding Switches to a Map](#) sections for more tips on arranging the map.

Monitoring the Network

Once the map has been arranged and saved, you can switch it to monitoring mode. You may already have noticed that devices were changing colors while you were arranging the map. This shows that *InterMapper* was already polling devices, even while you were editing the map's layout.

To change a map to monitoring mode, click the pencil icon at the lower left corner of the window, or press **Tab** as a keyboard shortcut. The pencil icon will change to have a slash through it, indicating that the map is no longer editable.



Figure 1-10: Monitoring mode indicators. The slash through the pencil indicates the map is not editable.

InterMapper has several indications to show the state of the network. First, icons of the devices change color to indicate different traffic and error rates. These are the states and default colors:

Normal: traffic and errors for the device are at normal rates (green)

Warning: traffic or errors have increased to an intermediate rate (yellow)

Alarm: traffic or errors have exceeded the alarm threshold (orange)

Down: the device no longer responds to *InterMapper* (flashing red)

Acknowledged: the device is down, but has been acknowledged by the operator (blue)

Unknown: the device is not being polled (gray)

When a device goes down, its icon will flash red. This flashing attracts attention to a new problem, but is distracting when left for a long time. To stop this flashing, you may *acknowledge* a failure to indicate that repairs are in progress. To do this, select one or more devices, then select **Acknowledge** from the **Network** menu. You will also be prompted to enter a text explanation for any actions you are taking, and then the devices will change to have blue icons.

InterMapper can also show links that are not operational. A link that has been taken down administratively appears with a blue X; a link that has failed is shown with a red X.



Figure 1-11: Non-operational links are shown with X's across the link

You may not want to see all the interfaces on a device. This is especially true as switches and hubs with dozens (or even hundreds) of interfaces become popular. You can hide them (choose **Hide Selection** from the **Edit** menu) so they are no longer visible. Deleting those interfaces and/or links will get rid of them temporarily, but they will re-appear when the map is opened again or when *InterMapper* re-scans that device.

Traffic flows which exceed a certain threshold are shown as dashed lines (which look like ants) traveling along links. Figure 9 shows a typical device. Here is a summary of the indications:

- short, widely spaced dashes indicate modest packet rates
- short, closely spaced dashes indicate high packet rates
- long dashed lines indicate long average packet length
- small circles indicate high error rates for an interface
- *halos* along a link indicate utilization greater than 50% (yellow) or 90% (orange)

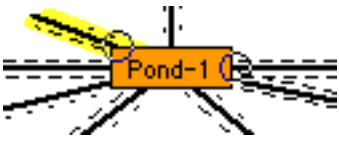


Figure 1-12: Traffic flows and error indications.

To set the error thresholds, use the **Device Thresholds...** choice in the **Network** menu. You may change the thresholds for the traffic indications with the **Traffic Thresholds...** command in the **Network** menu. The **Traffic Threshold...** window also displays a legend that describes the traffic indicators.

Pop-up Windows

InterMapper can display detailed information about an item on a map (a device, a network, or a link). Simply click and hold the mouse on an item -- a window will pop up displaying that item's current statistics. (If it doesn't pop up, click the pencil to place the map in monitoring mode.) You can also tear off this window by dragging the mouse outside its outline and releasing the mouse.

Device Pop-up Window

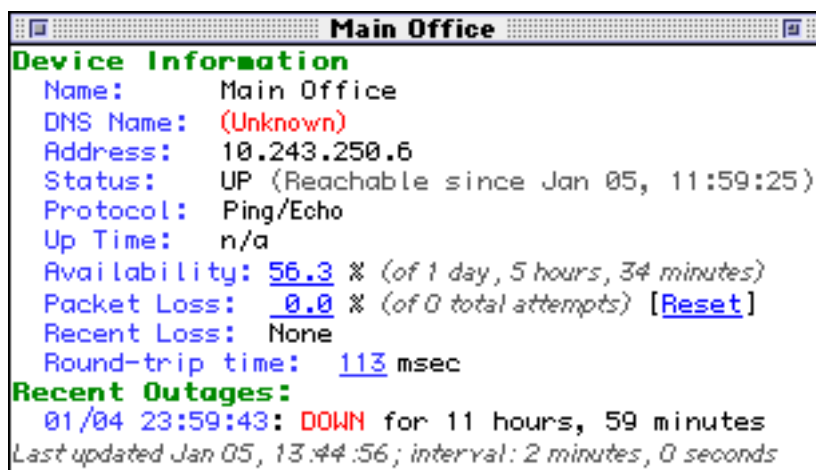
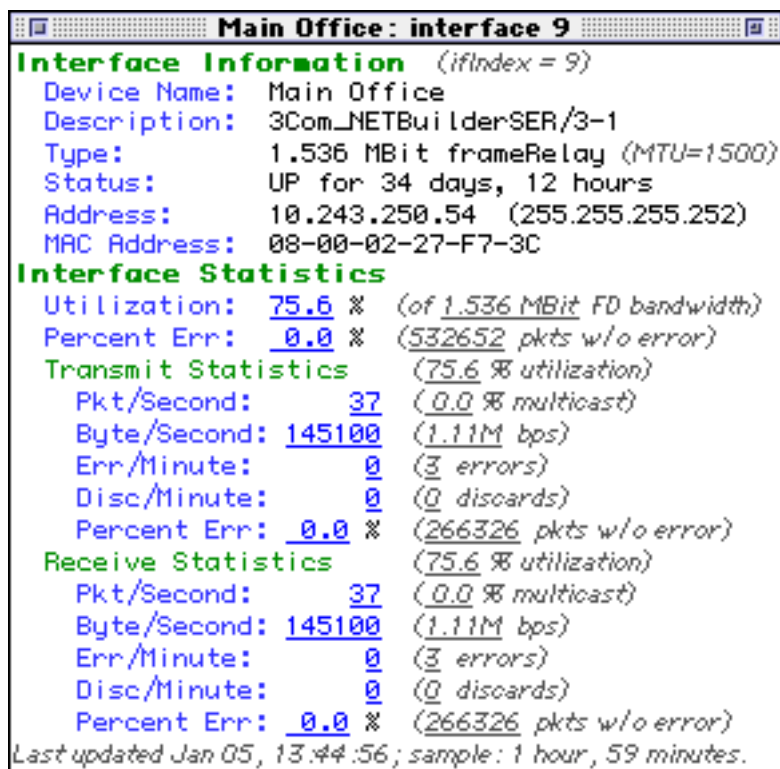


Figure 1-13: Device status pop-up window.

Clicking and holding the mouse on a device opens *device status* window that shows the device name, network address, its status, the protocol used to poll it, up-time (i.e., SNMP sysUptime, if available), availability (e.g., the percentage of the time the device was available based on the number of packets lost while testing), round-trip time (in msec), and spanning tree information (if available). When the device reports significant problems, the reason for the most important error will be shown in red at the bottom of the pop-up window. Figure 1-13 shows a device status pop-up window.

Tip: Click the underlined Reset link to set Packet Loss to zero. This will also reset the device's availability measurement.

Link Pop-up Window



The *link status* window shows the link's interface name and description, its type (10 or 100 Mbps, 1.5 Mbps T-1, etc.), its status and up-time, its IP, AppleTalk, and MAC addresses (when available), and traffic statistics (transmitted from and received by the interface), and the time since the last poll.

Tip: Certain devices do not report their link speed accurately in their SNMP responses. This will cause *InterMapper* to report a value which is not actually correct. You can work around this by clicking the underlined "bandwidth" link to open a window that allows you to set the link speed to the correct value.

Figure 1-14: Link statistics pop-up window.

Network Popup Window

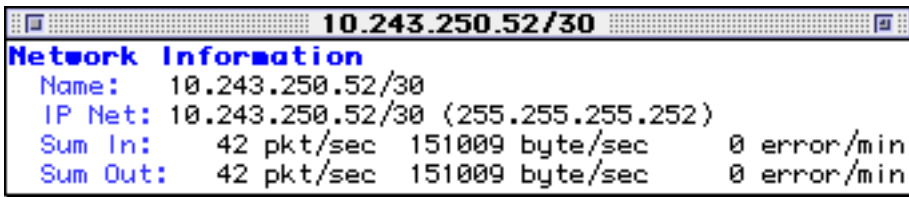


Figure 1-15: Network status pop-up window.

The *network status* window shows the network's IP address and subnet mask, AppleTalk address (if available) and a summary of the traffic being sent into and received out of the network.

Creating Strip Charts

InterMapper strip charts display the history of one or more variables. Strip charts are shown in floating windows that are revealed when the corresponding map is in front. This information can also be saved to a log file for further analysis.

To create a strip chart, open one of the pop-up windows, as described in the previous section, then tear it off to create a floating window. Click on any of the underlined values to create a new strip chart. To add variables to a chart, drag other underlined values to it. Up to twelve (12) variables may be placed in a strip chart. Figure 1-16 shows a typical strip chart.

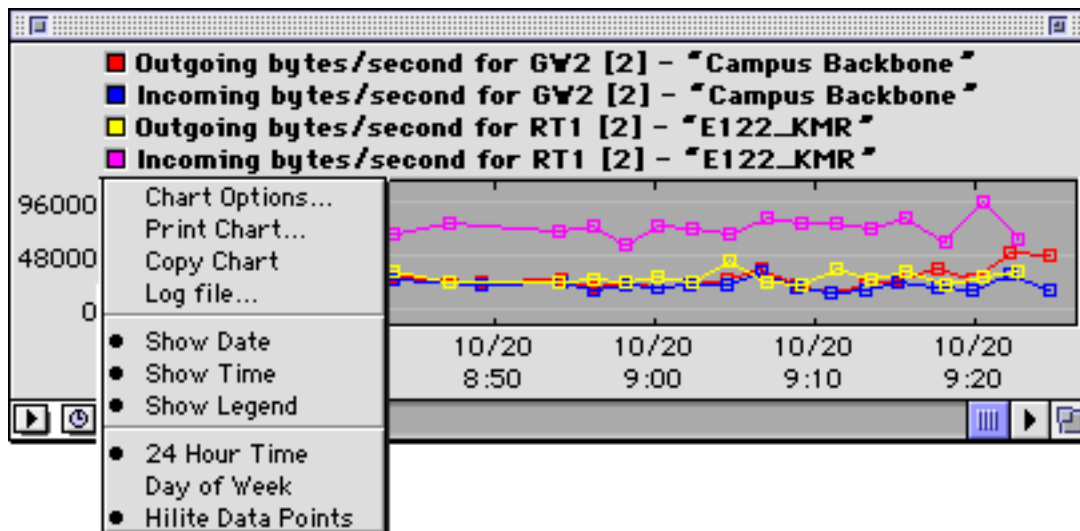
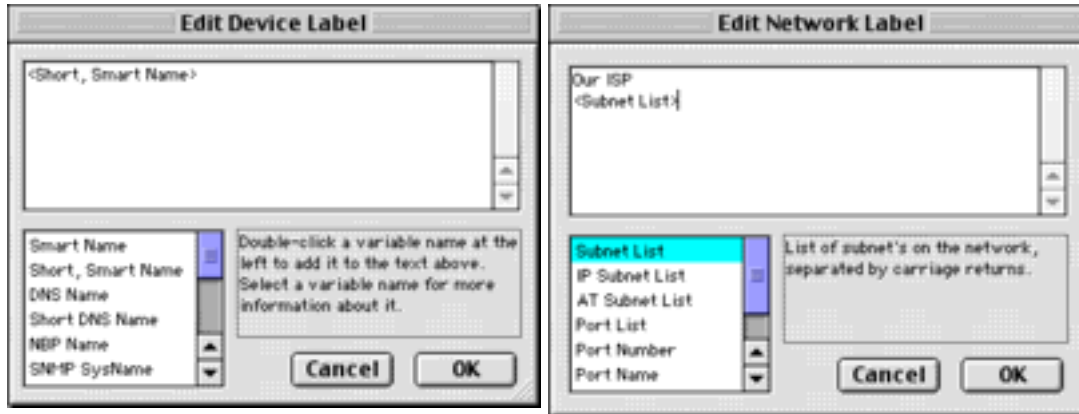


Figure 1-16: A strip chart showing four traces.

For more information about strip charts, see the Strip Chart page in [Chapter 5](#).

Modifying Device and Network Labels

Every device and network (rectangle and oval, respectively) has its own descriptive label. *InterMapper* creates a default label showing the device's full DNS name or IP address, or the link's IP and AppleTalk address(es). You may edit these labels by selecting one or more devices (or links) and choosing Label... (Cmd-L). This opens the windows of Figure 1-18 or Figure 1-19:



Figures 1-18 and 1-19: Editing Device and Network labels.

Whatever text you type in the window becomes part of the label of each selected item. Double clicking on any of the variable names in the list at the lower-left inserts that value into the item's label. For example, the devices in Figure 1-18, will be displayed with their short, smart name (i.e., the leftmost part of its full domain name). The network shown in Figure 1-19, will have a static (unchanging) label of "Our ISP" with a list of all the subnets in the network shown on the next line.

Scanning a Network

InterMapper can scan an IP or AppleTalk address range to discover all the devices on that network. It will then add those discovered devices to the map, and connect them to the proper network.

To scan a network for devices, double-click a network oval or select a network and choose **Get Info...** from the Edit menu. This will open the window shown in Figure 1-20, below.

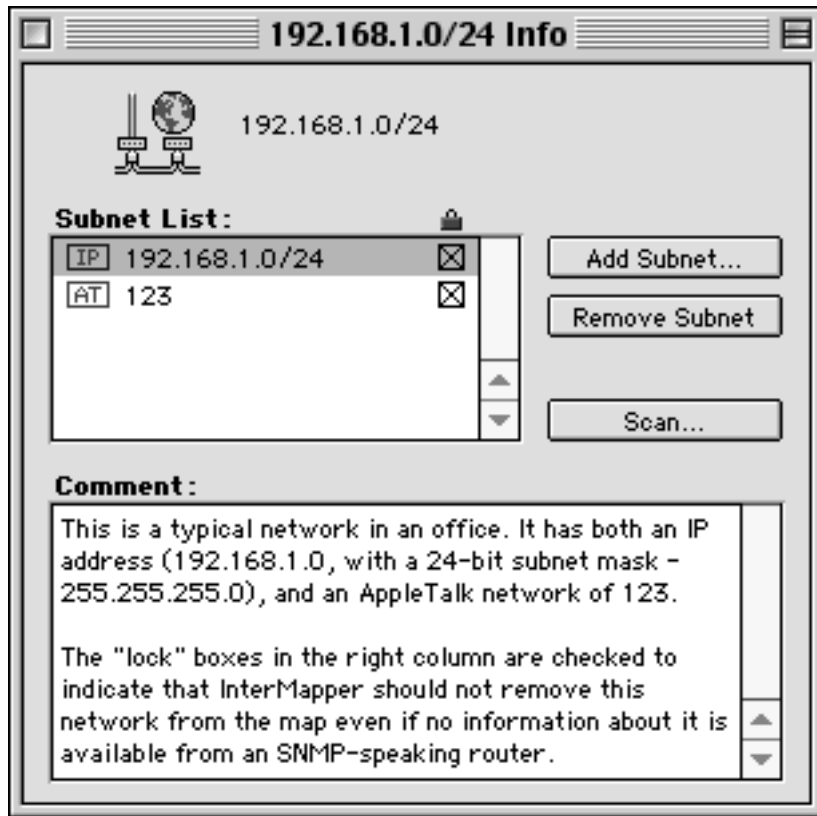


Figure 1-20: Network Information Window.

Select a network from the list at the upper left, and click the **Scan...** button. This will open a window similar to Figure 1-21.

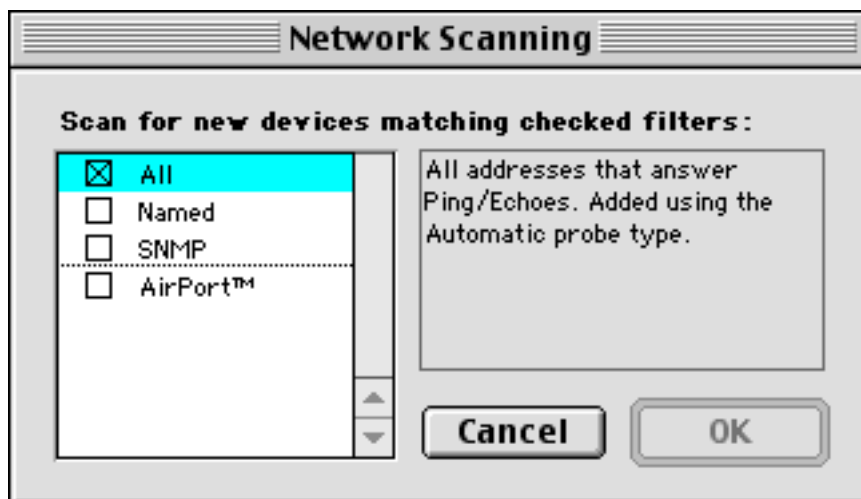


Figure 1-21: Autodiscovery Filters. The checkboxes indicate the kinds of devices you want to find.

Check the boxes that match the kind of devices you want to find. Click **OK** and scanning will begin.

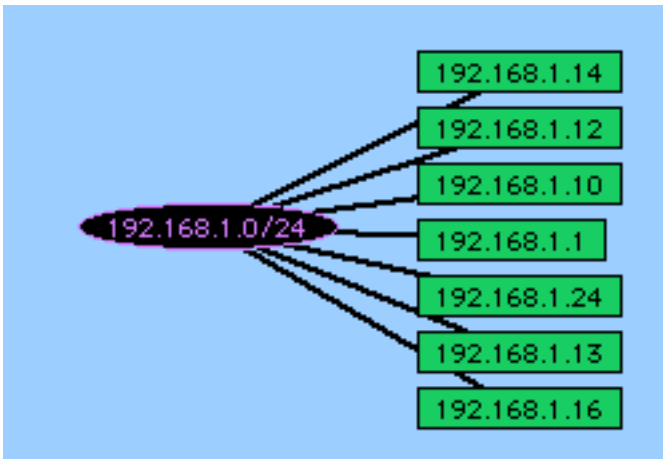


Figure 1-22: InterMapper scanning a network.

The oval itself will turn purple while the scan is continuing. When complete, the oval will revert to the default color.

Adding Networks to the Map

You can add new networks (*ovals*) to the map manually. This is convenient if InterMapper has not found all the subnetworks, perhaps because some of the routers don't speak SNMP.

To add a new network, choose **Add Network...** from the **Network** menu. A window like Figure 1-23 will appear. Enter the IP subnet information or the AppleTalk net number or range and click **OK**.

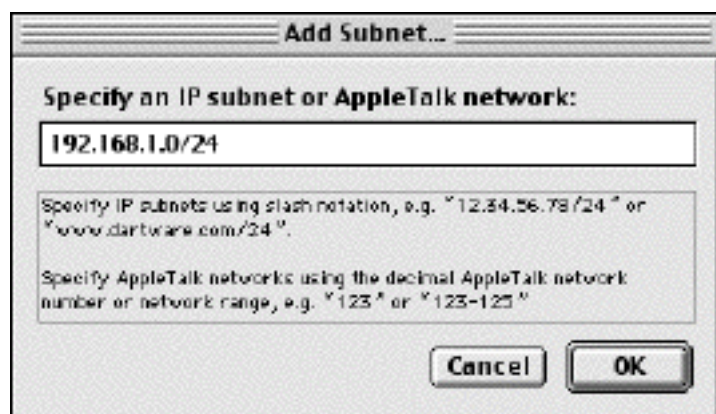


Figure 1-23: *Add Subnet...* window. Enter an IP subnet (in the form *x.x.x.x/yy*), an AppleTalk network number, or an AppleTalk network range (*###-###*).

The network will be added to the map as an oval, labelled with the specified network information. Devices that belong to that network will automatically connect themselves to the new network.

Adding a Network to an Existing Network

Some physical links have two or more separate subnets present on the same "wire" (for example, two IP subnets, or an AppleTalk network range to an existing subnet). To add another network to an existing network oval, double-click the network. You'll see the Network Information window shown in Figure 1-24. Click the **Add Subnet...** button and enter the new network information as described above in Figure 1-23.

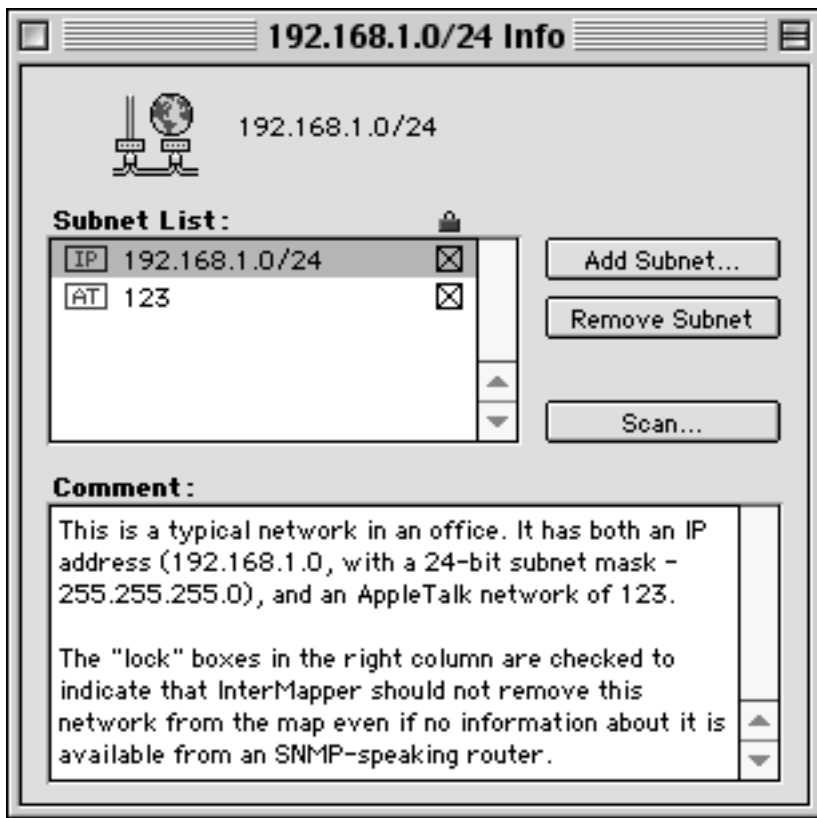


Figure 1-24: Network Information Window.

Making Connections Manually

InterMapper may not connect devices to the proper network in every case. To add a link manually, hold down the "E" key, and drag from the device to the network. This process is shown in Figure 1-25.

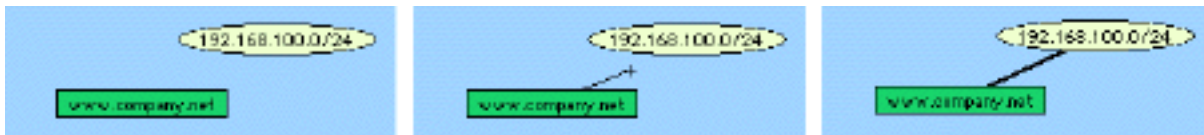


Figure 1-25: Manually adding a link between a device and a network. Hold down the **E** key while dragging between the two devices. Notice the link being dragged from the rectangle to the oval.

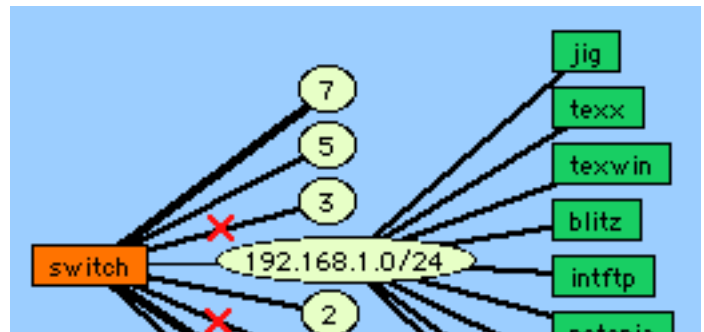
Once a manual connection has been established, InterMapper will remember it. You can drag the items around the map, and they will work just like those InterMapper has automatically created. To remove a manually-added link, click on the link, drag it away from the network, and hold down the **Delete** key before releasing the mouse button.

Connecting Devices to Switch Ports

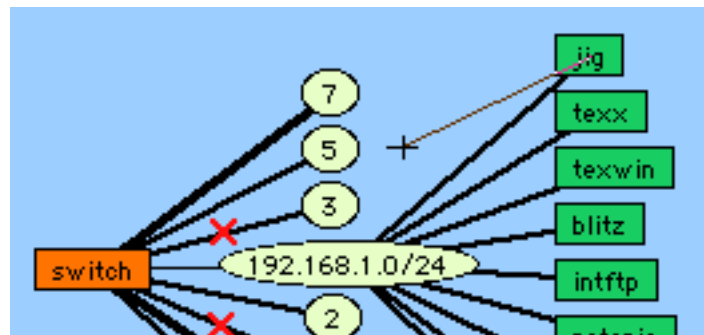
InterMapper does not connect devices to the proper port of a switch. Instead, *InterMapper* connects all the devices of a subnet to the first switch port it discovers (usually the port with `ifIndex=1`).

You can manually connect devices to the proper ports by dragging the link from the central oval (labeled "192.168.1.0/24" in the figures below) to the proper port. This is shown in the sequence of graphics below:

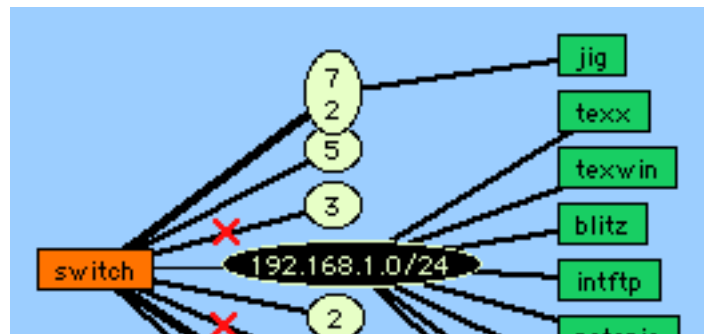
1. The map before making changes. The switch's ports are shown by the numbered ovals. (Make sure the map is in editing mode.)



2. Click on a link and drag it. You'll see a line follow the cursor.



3. Drag the link to the desired port. It will disconnect from the original network oval and remain connected to the new. Note that the port's oval now contains two port numbers: that of the switch (7) and the port number of the device (2).



Tip: When moving links to the proper port on a switch, it's sometimes easier to change the port labels to display the port's number. To do this, select all the ports, and then choose **Label...** (Cmd-L) from the **Network** menu. Double-click on the **<Port Number>** variable, as shown in [Figure 1-18](#). The oval (network) labels will display the port numbers.

Background Images

InterMapper allows you to place a background image on a map so that it appears behind the map contents - the devices, icons, and links on the map. These images might show the locations of equipment in an office or against a map of a city or country. The two figures below show a map before and after placing an image in the background.

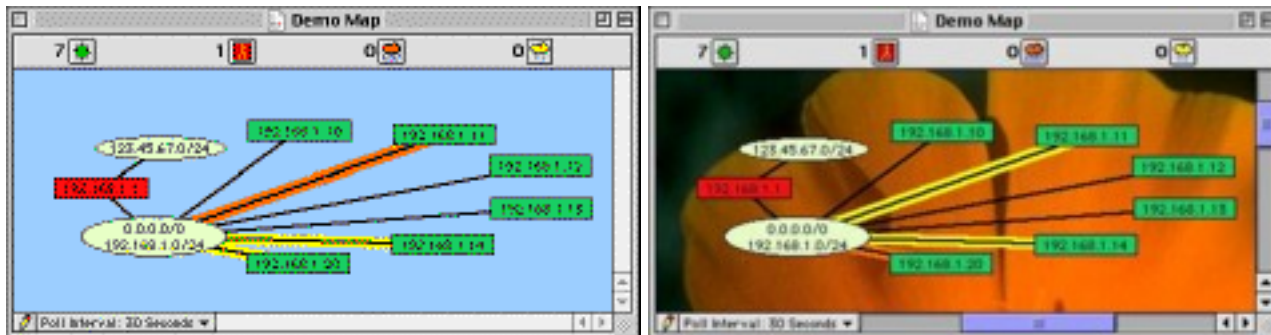


Figure 4-16a: Map without background image. Figure 4-16b: Same map with background image.

For details of placing a background image on the map, see the [Preferences](#) chapter.

Using Helper Programs

InterMapper can use helper programs to create maps or to troubleshoot problems. These programs are available through a contextual pop-up menu. To invoke a helper program, command-click on a device or network. You will see the following pop-up menus:

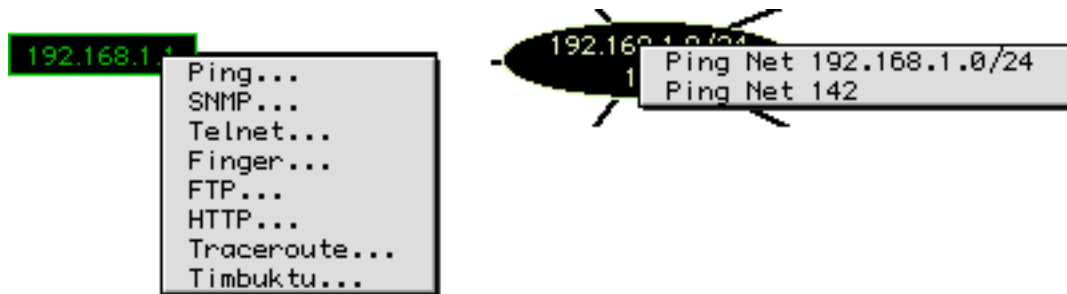


Figure 1-28: The Helper Program pop-up menus. Command-clicking on a device results in the left pop-up menu; command-clicking on a network results in the right pop-up.

Selecting one of the items in the Helper Program pop-up menu launches an application using the device as its target. For Classic MacOS, *InterMapper* uses the facilities of the Internet control panel (or Internet Config for systems before MacOS 8.0) to launch the proper viewer for http, telnet, ftp, etc.

Ping... Launches the "ping" utility specified with Internet Config (Classic MacOS)

SNMP... Launches *SNMP Watcher*, an SNMP query tool

Telnet... Launches the Telnet utility

Finger... Launches the Finger utility

FTP... Launches the FTP utility

HTTP... Launches the Web browser

Traceroute... Launches the traceroute utility

Timbaktu... Launches the Timbaktu application

In MacOS X, *InterMapper* launches the Terminal application for the following helpers:

Ping... Launches the Terminal application, running the ping program for the specified address

SNMP... Launches *SNMP Watcher*, an SNMP query tool

Telnet... Launches the Terminal application

FTP... Launches the default FTP tool

HTTP... Launches the default Web browser

Traceroute... Launches the Terminal application, running the traceroute utility

Timbaktu... Launches the Timbaktu application

Here are some tips for using auxiliary programs with *InterMapper*:

When building a map, you can use *MacPing* to discover devices on a particular AppleTalk or IP network. For AppleTalk, simply choose the AppleTalk net desired. For IP, choose **Test IP Devices...** and enter an IP address or range of addresses. *MacPing* will create a list of all the devices that respond. Select the desired devices from the *MacPing* window, and choose **Copy**. Then paste those devices into the "Add Device(s)..." window.

MacPing is also a good test tool when troubleshooting problems with multiple devices a network. As shown in Figure 1-28, you may command-click on a device, then select **Ping...** *InterMapper* sends an AppleEvent to *MacPing* to begin testing that device. Command clicking a network in an *InterMapper* map will instruct *MacPing* to test the devices in that subnet. A demonstration version of *MacPing* is available at <http://www.macping.com/>

InterMapper can also launch *SNMP Watcher* to query a device's MIB variables. To do this, command-click on a device, and choose **SNMP...** *SNMP Watcher* can send SNMP queries to a device and can also parse proprietary MIB's. A demonstration version of *SNMP Watcher* is available at <http://www.snmpwatcher.com/>

On systems older than MacOS 8.0, you can use *Internet Config* to specify a default Web browser or Telnet, Finger, FTP, Ping, or traceroute utility that will be launched. *Internet Config* functionality is built into MacOS 8.0 and above; for earlier systems, the original *Internet Config* software from Peter Lewis and Quinn is available by searching the VersionTracker site at <http://www.versiontracker.com>.

Command and Menu Reference

This chapter describes each of the menu commands in detail. As with all Macintosh programs, the menus contain the commands that make the program operate. The menus are:

[Apple Menu](#)

The **Apple** menu has the **About...** box, which describes InterMapper, and the **Register InterMapper...** command

[File Menu](#)

The **File** menu contains commands for opening, closing, and saving files, for printing windows, and for quitting InterMapper.

[Edit Menu](#)

The **Edit** menu contains commands for copying and pasting data as well as commands for selecting and hiding items in maps.

[Network Menu](#)

The **Network** menu contains commands for adding devices and networks to a map, changing information about those items, and setting and acknowledging notifications of alerts.

[Layout Menu](#)

The **Layout** menu contains commands for arranging items on the map.

[Display Menu](#)

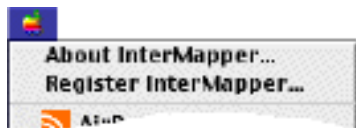
The **Display** menu contains commands for changing the appearance of items on the map.

[Window Menu](#)

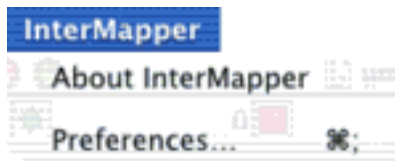
The **Window** menu contains selecting which windows are visible.

Note: Certain menu items have moved between the Classic and MacOS X versions. For example, the **Preferences** command is in the **Edit** menu in Classic MacOS, while it's in the **InterMapper** menu for MacOS X. We have placed menu items in the same place where possible, and where consistent with the style of the underlying operating system.

Apple Menu



The Apple menu for Classic MacOS contains **About InterMapper...** and the **Register InterMapper...** commands.



The InterMapper menu for MacOS X contains these commands, as well as the **Preferences...** command and the standard commands for hiding and quitting InterMapper.

The **About...** box contains the *InterMapper* copyright notice, version number, and credits. Clicking on any of the underlined phrases will launch your preferred e-mail program or web browser. Figures 2-1, 2-2, and 2-3 show the About box.



Figure 2-1: The *InterMapper* About box.



Figure 2-2: The *InterMapper* Credits pane.

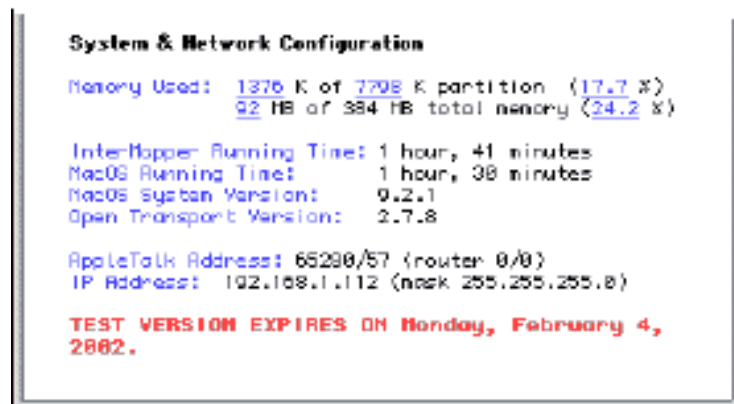


Figure 2-3: The *InterMapper* Statistics pane.

The *Credits Pane* of the About box shows credits and the copyright notice.

The *Statistics Pane* of the About box (click the right or left arrow) displays information about the program, including the CPU and memory usage, running time, whether Open Transport or MacTCP is currently in use, and whether the FAT, native, or 68K version of *InterMapper* is running. (*Mac OS X*: There are no memory usage statistics displayed.)

Register InterMapper...: InterMapper requires a serial number to unlock a demonstration version so that it

runs as the full version. The Registration window is shown in Figure 2-4.



Figure 2-4: Registration window. Enter your name and serial number as specified in the registration message.

When you purchase InterMapper, you will receive a registered name and serial number. Enter these into the window shown in Figure 2-4.

About Serial Numbers

InterMapper supports a number of different serial number formats. A full serial number - sent to those who purchase the software - will never time out. Once entered, that version of the application will run forever.

InterMapper also supports evaluation serial numbers which will operate for a certain number of days before it ceases to operate. This gives you an opportunity to try the program without obligation. When a serial number times out, InterMapper simply ceases to operate. It will *never* delete or alter files on your hard drive.

File Menu



The **File** menu holds the standard Macintosh New, Open, Close, Save, and Print commands.

New: (Cmd-N) Creates a new map. Creating a new map opens the Autodiscovery window that allows you to specify the starting point for auto-discovery. See the [Autodiscovery](#) menu command for more information.

Open: (Cmd-O) Opens a saved map. *InterMapper* will open the selected map and begin polling the devices contained in it. A device that appears in multiple maps will be polled once for each map.

Close: (Cmd-W) Closes a map and stops polling the devices contained in that map. Inadvertently closing a map inadvertently could cause a loss of oversight in your network: therefore, *InterMapper* presents a warning dialog, asking if you really want to close the map.

Tip: This "Warn before closing" behavior can be disabled with a [preference](#).

Save: (Cmd-S) Saves the current state of the map. *InterMapper* saves complete information about the map, including the location and shape (icon/rectangle/etc.) for each device and network. *InterMapper* also saves a list of devices that are currently acknowledged. These devices will remain acknowledged after the map is reopened.

Tip: If you save a map in the same folder as the *InterMapper* application, it will be opened automatically the next time *InterMapper* is launched. This behavior can be disabled with a [preference](#).

Tip: To save all open map windows at once, hold down the option key while choosing the Save command.

Save As...: Saves the current map by a different name, or in a different folder or drive.

Export As: Save the information save the map as a PICT, as an EtherPeek name table, or as a tab-delimited text file (which can easily be imported into a spreadsheet program such as Microsoft Excel.) The three formats are described below:

- A PICT file, the native graphical file format for the Macintosh. These documents may be opened by the *SimpleText* application.
- An [EtherPeek](#) Names table. This is a tab-separated list of names and addresses with the following columns:

```
DNSname [tab] protocol [tab] MAC/IP address where available
```

- *InterMapper* can export information about a map in a tab-delimited file format, suitable for opening with Excel or other spreadsheet program. For devices, the columns are:

```
Status User-Defined Name Address Probe Type DNS/NBP Name sysUpTime sysName
sysDescr sysContact sysLocation Comment User List
```

The column heads for links are:

```
Status Device Name ID ifDescr ifSpeed ifMtu Addresses ifLastChange PhysicalAddress
```

Page Setup: *InterMapper* supports the standard Macintosh Page Setup dialog.

Tip: Hold down Option when choosing Page Setup. This will provide instructions for printing the entire map on a single page.

Print: (Cmd-P) *InterMapper* will print the current window on the currently selected printer. It will use as many pages as necessary to print the entire map or window contents.

Quit: (Cmd-Q) Quits the application. *InterMapper* will present a warning dialog to ensure that you really want to quit the application. (*MacOS X:* The Quit menu command is in the **InterMapper** menu.)

Tip: This "Warn before quitting" behavior can be disabled with a [preference](#).

Edit Menu

Edit	
Undo	⌘Z
Cut	⌘X
Copy	⌘C
Paste	⌘V
Clear	⌘-
Hide Selection	⇧⌘H
Show Adjacent	⌘Y
Show All	
Select All	⌘A
Select Adjacent	⌘J
Select Other	▶
Find...	⌘F
Find Again	⌘G
Sound	
<input checked="" type="checkbox"/> Map Editor	⇧⌘I
Map Settings...	
Preferences...	

The Edit menu contains standard Macintosh editing commands, as well as various commands for selecting and finding items.

Undo: Most operations in *InterMapper* can be undone. Undo is one level deep: choosing Undo a second time will redo the operation.

Cut: (Cmd-X) These four commands work as with most other
Copy: (Cmd-C) Macintosh applications.
Paste: (Cmd-V)
Clear: (Cmd- hyphen)

Hide Selection: (Cmd-Shift-H) Hide the selected item(s). Those items will not be visible in the map, nor will they be counted as down. Certain devices, such as switches may have ports that are not in use. *InterMapper* will, by default, show those unused interfaces as down. Hiding those interfaces avoids the false-down situation.

Show Adjacent: (Cmd-Y) Make visible all the hidden links and nodes connected to the selection.

Show All: Makes all the hidden items visible in the current map.

Select All: (Cmd-A) Select all the devices in a map or all the text in a text field.

Select Adjacent: (Cmd-J and option-clicking are keyboard shortcuts.) Selects all the devices and networks that are adjacent to (one level out from) the current set of selected items.

Select Other is a hierarchical menu for making other detailed selections:

All devices: Selects all devices (but not links or networks)

Down devices: Selects only the devices that are currently marked as down.

Up devices: Selects only the devices that are currently marked as up.

All networks: Selects all networks, but not the devices.

Network size...: Selects all networks with exactly that number of devices connected to it.

Unselected: Inverts the selection. Items that are selected will not be; items that are not selected become so.

Find...: (Cmd-F) Opens a dialog where you can enter a string to find. When a device is found, its map is moved to the front and the device is highlighted with zooming rectangles.

Find Again: (Cmd-G) Finds the next item that matches the specified string.

Sound: When checked, sounds *InterMapper* produces will be audible through the computer's speaker.

Map Editor: (Pressing the Tab key, or clicking on the pencil icon at the lower left of the map are keyboard shortcuts.) Toggles the map between editing mode (where the map may be rearranged, edited, and changed) and monitoring mode (where the map reports displays the current state of the network.)

Map Settings...: You can control an individual map's color settings as well as access to its web pages. See the [Map Colors](#) page.

Preferences...: There are many categories of preferences. See the [InterMapper Preferences](#) chapter for more information. (MacOS X: The Preferences is part of the **InterMapper** menu.)

Network Menu



The **Network** menu contains commands to add devices or networks, or to modify information about existing items in the map. In addition, the **Acknowledge** command indicates that a device is known to be down, but is being worked on.

Auto-Discover...: (Cmd-Shift-+) InterMapper will automatically find network devices such as routers, hosts, switches, hubs, servers, workstations, etc. and place them on the map. This command allows you to control the kinds of devices InterMapper will find and the breadth of its search.

InterMapper uses a *starting address* and then scans the for additional devices. By default, InterMapper uses its router's address or its own IP address. However, you may enter a different address or DNS name as a starting point. If InterMapper finds SNMP-speaking routers with connections on other networks, it will search those networks, hop-by-hop, finding more devices (and possibly more routers) until the

hop limit is reached.

The **Autodiscovery** window shown in Figure 2-5 allows you to specify the starting address as well as specifying other options for the autodiscovery process.



Figure 2-5: The Autodiscovery dialog.

The auto-discovery process also allows you to select which kinds of devices are to be added to the map. InterMapper applies a set of *filters* to the discovered devices. Only those that match the checked filters will be added to the map.

Starting host name: This is the name or address of a device that InterMapper should use to begin the autodiscovery process.

Specify a SNMP community string: allows you to specify an additional SNMP Read-only community string to be used to interrogate all devices. (InterMapper always attempts to read SNMP information using the default 'public' community string.)

Stay within __ hops of starting device stops autodiscovery after InterMapper has searched the specified number of hops from the starting device.

Scan for devices on all networks is discussed below.

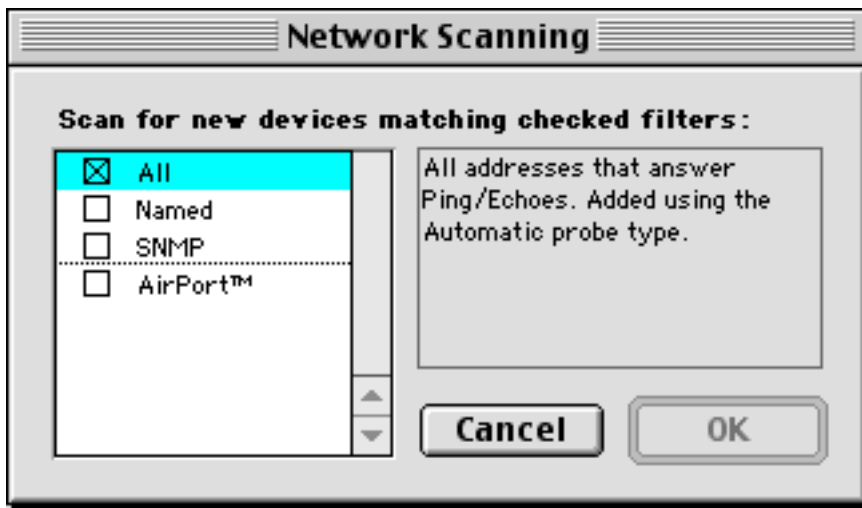


Figure 2-6: The Network Scanning dialog. Check a box to seek the associated device.

Clicking the **Filter...** button of Figure 2-5 opens the **Network Scanning** window shown in Figure 2-6. These options are:

- **All** forces a complete IP address scan for each network. InterMapper sends an ICMP Ping request to each IP address in the subnet range.
- **Named** Each IP address in the subnet is looked up in the DNS. If a corresponding name is present, the device is added to the map
- **SNMP** InterMapper sends a SNMP GetRequest to each address in the range. Devices that respond are added to the map.
- **AirPort** If the SNMP response indicates it's an Apple AirPort Base Station, the device is added to the map.

Warning: The **All** checkbox will discover everything on a network. On a small or medium-sized network, this might be a reasonable option to choose. On large networks, the **All** choice may discover far too many devices to make a workable map.

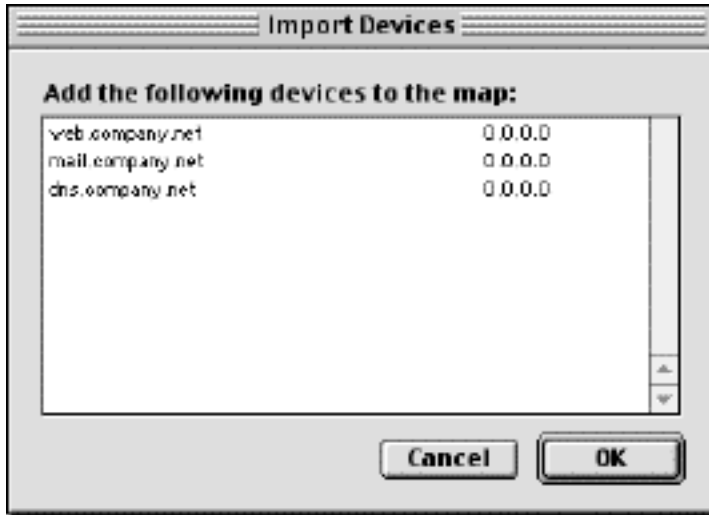
Add Device(s) (Cmd-+): Adds new devices to a map. During autodiscovery, *InterMapper* will draw lines that connect the newly-discovered device(s) to networks already in the map. Figure 2-7 shows the **Add Device(s)** window.

To use the **Add Device(s)...** command, enter the one or more device names or addresses into the window. These names may be typed manually or the names may be pasted from some other source. The names must be separated with commas or whitespace (spaces, tabs, or returns). The list of host names or IP addresses might come from a text file, be copied from the [MacPing](#) program or a traceroute program, or from other source of names and/or addresses.



Figure 2-7: Add Device(s) window.

You may also specify a probe type that will be used to test the device. *Automatic* tells InterMapper to use SNMP or ICMP Echo for IP devices; SNMP or AppleTalk Echo for AppleTalk devices. You may also specify probes for web servers, mail servers, or any of the other probes shown in the popup menu. See [Appendix A - Probe Reference](#) for a full listing of the built-in probes.



Tip: It is also possible to import devices into a map by pasting DNS names or IP addresses. Simply copy a list of device names or addresses, one per line, from a text document, click on a map, and choose **Paste**. A window like Figure 2-7a will appear. Click OK, and the items will be added to the map.

Figure 2-7a: Importing devices by pasting device names and addresses directly into the map.

Add Network: Add a network (oval) to the map. This is useful when *InterMapper* does not automatically detect the network because no SNMP-speaking devices are present.

The window shown in Figure 2-8 will appear. Enter the IP or AppleTalk network information. (For a discussion of how IP network information is represented, along with a discussion of the "/24" etc notation, see [Subnet Mask FAQ](#).)

After you click **OK**, you will see a new network oval on the map representing that subnet. You can connect devices to this network by dragging their links as described in [Manual Connections](#) page.

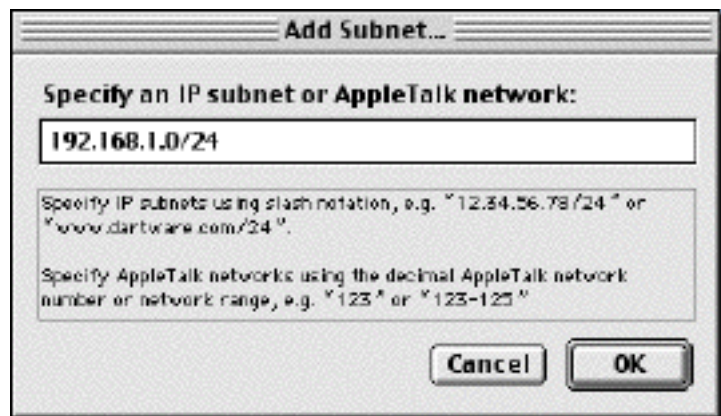


Figure 2-8: Add Subnet... window. Enter an IP network or an AppleTalk network number range.

Refresh Map: (Cmd-K) Causes *InterMapper* to re-poll the selected device (or devices). This is useful for retrieving the status of a device or detecting that it has returned to service. If a single device is selected, it is polled as soon as possible. If many devices are selected, they are moved to the head of the poll queue so they will be polled as soon as possible.

Get Info: (Cmd-I) Opens a window that contains a compact summary of information about the selected devices or networks. Read the [Device Information Window](#) and [Network Information Window](#) sections of this manual for more information.

Set Info: Allows you to set parameters on multiple items at a single time.

- **Set Community...** sets the [read-only community string](#) for all selected devices. Figure 2-9 shows a typical **Set Community...** window. The default community string for most SNMP devices is "public".

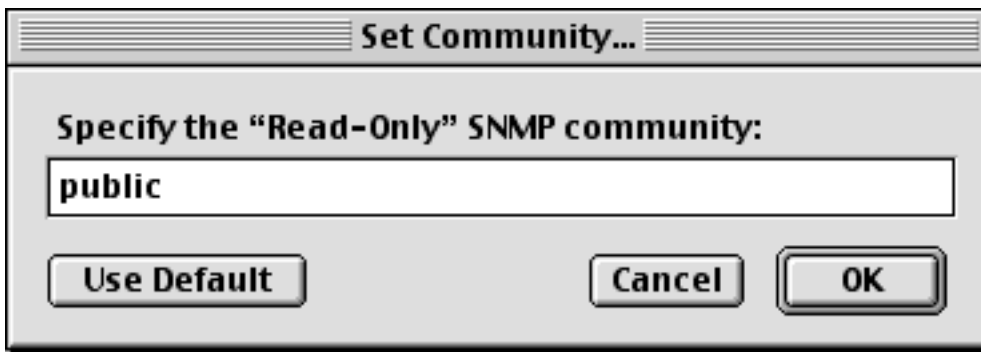


Figure 2-9: Setting the Read-Only SNMP Community string.

- **Set Timeout...** sets the probe timeout for all selected devices. This opens the window shown in Figure 2-10. The default timeout for InterMapper is 3 seconds. See the [Probe Reference](#) chapter for details on the probe timeouts.

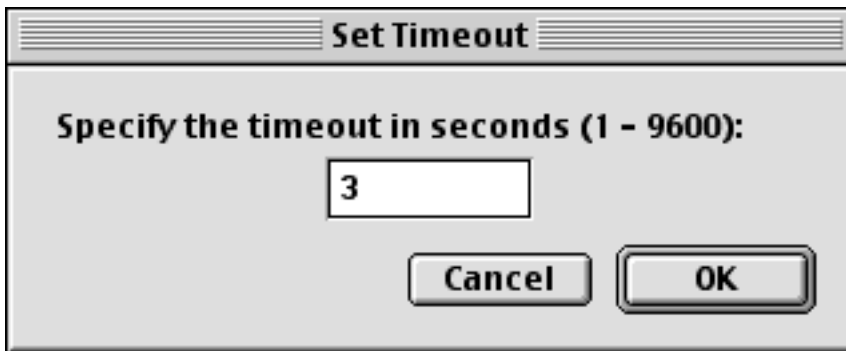


Figure 2-10: Setting the probe timeout.

- **Set Comment...** sets the comment for all the selected devices. (See the [Device information window](#) for details on the comment field.)

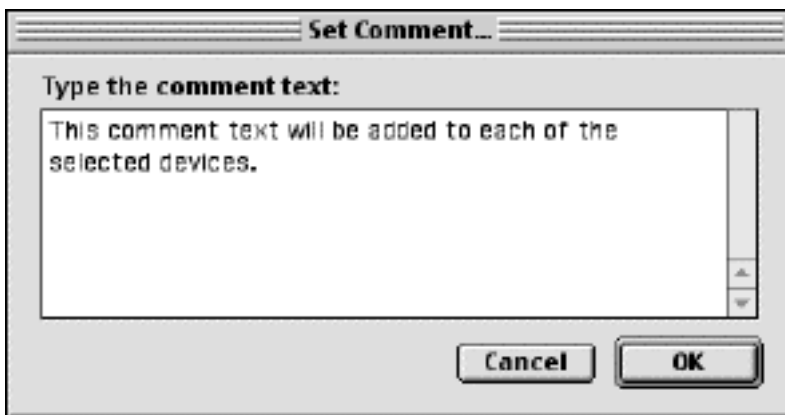


Figure 2-10: Setting the comment for devices.

Attach Notification...: Every device can send notifications when it changes state. Use this command to select which notifications will be sent when the device changes to a particular state. Figure 2-11 shows a typical set of notifications. Read the [Notifications](#) chapter for a complete description of this window.

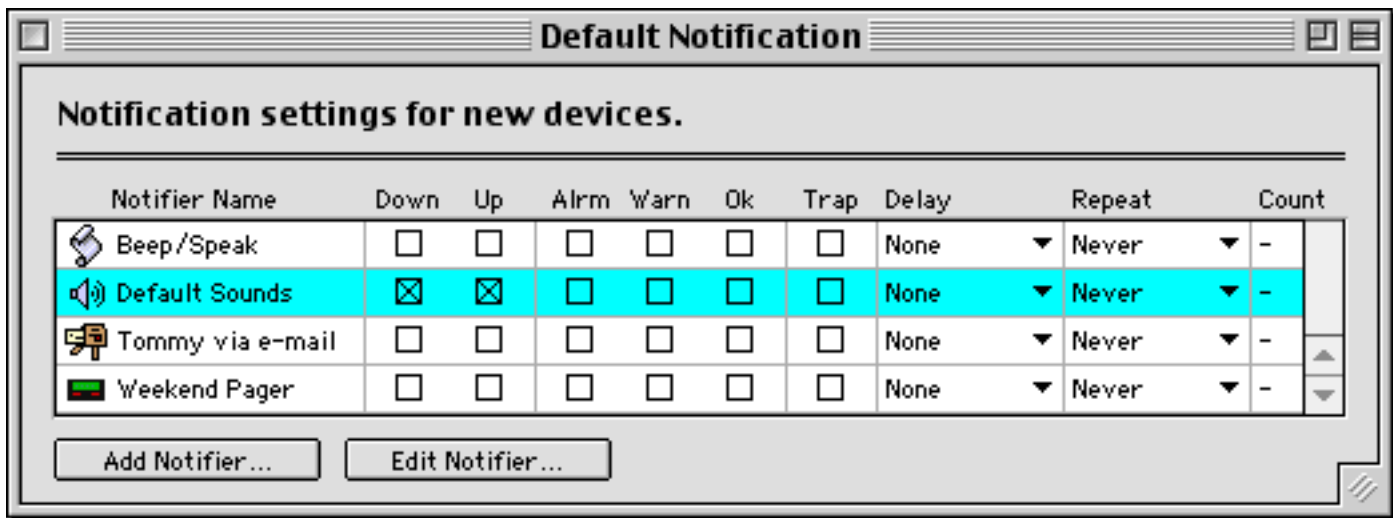


Figure 2-11: Selecting notifications. Check a box to indicate that a particular notification should be sent when the device changes to the checked state.

Default notification...: Use this to set the notifications that will be attached to all new devices (but not existing devices on the map). Figure 2-11 above shows a typical set of notifications.

Device Thresholds...: Set the criteria for indicating that a device is down, in alarm, or in warning. These settings apply to all the devices of a single map. Note that these are default settings. They may be overridden by the thresholds of a specific probe.

- **Down:** this is the most serious condition. It means the device no longer is responding to probes. The parameter is the number of packets that may be lost before declaring the device down.

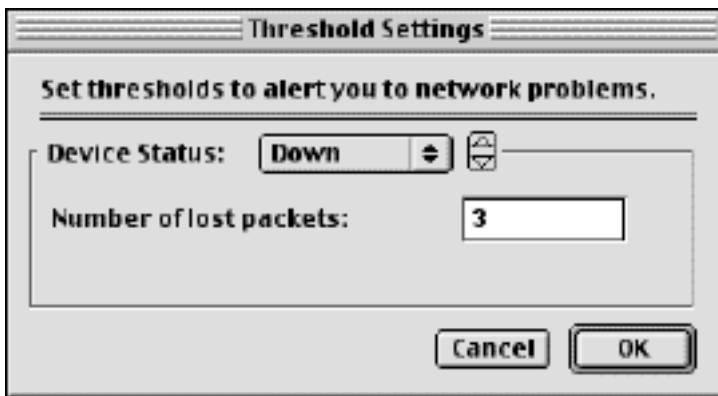


Figure 2-12: Device Threshold window.

- **Alarm:** this is next most serious condition. The parameter is the number of interface errors (per minute) allowed before marking the device in alarm.

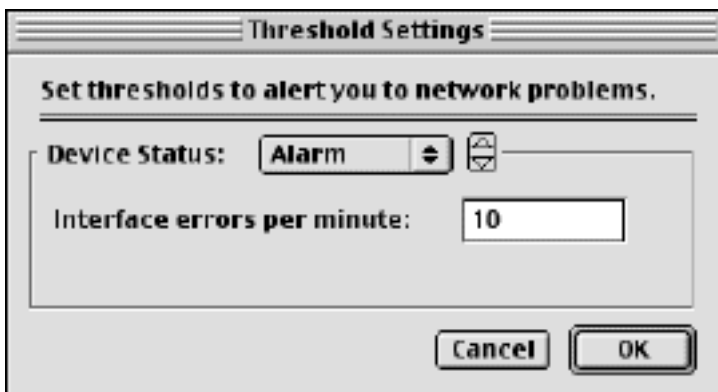


Figure 2-13: Setting the device Alarm threshold.

- **Warning:** the least serious error state. The parameter is the number of interface errors (per minute) allowed before showing the device in warning.

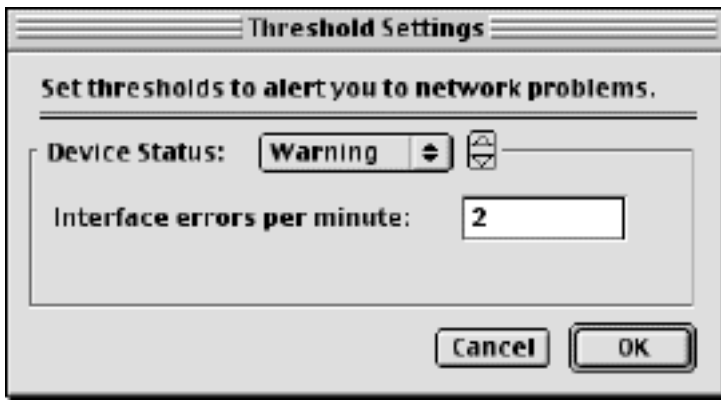


Figure 2-14: Setting the device Warning threshold.

Traffic Thresholds...: Use this command to set the levels at which the traffic indicators (often called *ants*) are displayed on the links. These settings apply to all links on a map. There are different patterns to indicate different volumes of traffic. The figure below shows the thresholds and the on-screen indications.

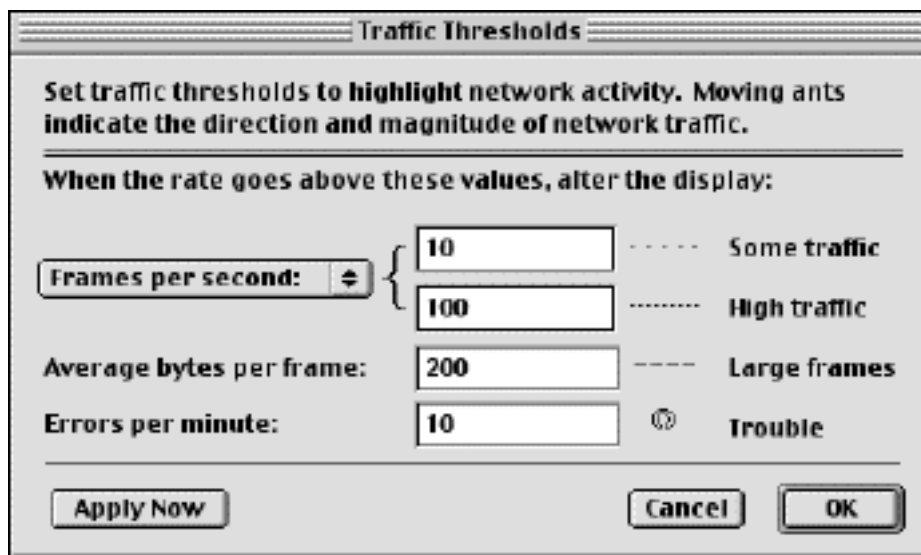


Figure 2-15: Thresholds for traffic (the "ants") can be set using this window.

There are three configurable traffic level indicators. The short, widely spaced dashes indicate a minimal level of traffic, above than the first threshold. The short, closely-spaced dashes indicate a higher level of traffic, exceeding the second threshold. Longer dashes indicate that the average frame length is greater than the specified threshold. The frame length indication takes precedence over the high traffic indication.

Finally, small circles (resembling raindrops in a puddle) at the ends of links indicate an error rate above the designated threshold on that interface. If the circles appear on the link next to the device, it signifies receive errors for that interface. Circles next to the network represent transmit errors for the interface.

Acknowledge: (Cmd-') A device that is down is indicated by a flashing red icon. This serves to attract attention, but can be distracting after corrective action has been initiated. *InterMapper* allows you to *acknowledge* the failure to eliminate this flashing icon. Acknowledged items will turn blue. In addition, acknowledging a device turns off recurring notifications.

When the **Acknowledge** command is selected, *InterMapper* will prompt the operator for a comment regarding the device. Figure 2-16 below shows the window for entering the message. This text will also be saved as an entry in *InterMapper's* log file.



Figure 2-16: The Acknowledge Message Window. Data typed here will be entered into the InterMapper log file as well. Checking the **Don't send notifications...** box will hold off notifications for the specified number of minutes.

Snooze Alarm

You may also check the **Don't send notifications for ___ minutes** box. (This is called the notification "snooze alarm".) InterMapper discards all notifications that are currently queued to be sent, as well as ignoring all notifications to be sent during the specified number of minutes. This is useful when a major device has failed, and can abort a flood of notifications.

Un-Acknowledge: This command restores the flashing icon for a device that has been acknowledged in error, or which needs further attention. Un-acknowledging a device reactivates recurring notifications.

Layout Menu



The **Layout** menu contains commands for changing the arrangement of items in a map. These commands offer powerful tools for moving many devices and links at one time. Read the **Tutorial** section of this manual for tips on arranging maps.

The following commands work identically for devices (usually shown as rectangles) and networks (ovals) unless otherwise specified.

Cycle: Move the selected items into an oval around the edge of the window. This allows you to see the interconnections between the devices of your network more easily. Figure 4 in the [Tutorial](#) illustrates the Cycle command's action.

Bus: Arrange items into a vertical column, changing the item that connects them into a vertical *bus* shape. This might represent a group of devices connected by an Ethernet or other broadcast medium. Figure 1-8 in the [Tutorial](#) illustrates the Bus command's action.

Star: Arrange items so they surround a network or device that connects them. The devices will be spaced equally around the circumference of a circle. Figure 1-9 in the [Tutorial](#) illustrates the Star command's action.

Align (Cmd-Shift-K): Align the selected items relative to each other. The **Align ...** buttons work like other drawing programs. The *Distribute* choice will space the devices evenly. If the *Across range* box is checked, the items will be distributed evenly in the space that the items occupy; if it is not checked, the items will be drawn with a small amount of space between the icons. Figure 2-17 shows the options for aligning items.

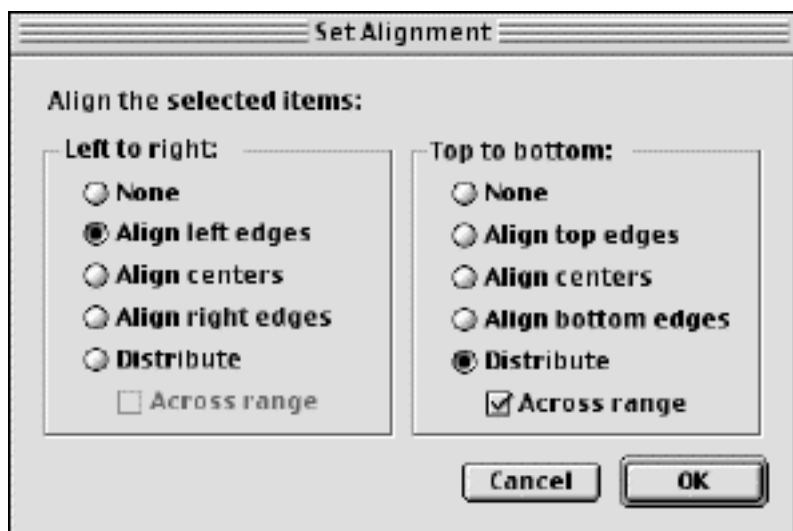


Figure 2-17: Changing alignment of selected items in a map.

Rotate: Rotate the positions (but not the text or icons) of the selected items as a group. Items will be rotated clockwise by the number of degrees specified. Figure 2-18 shows the window for rotating items.

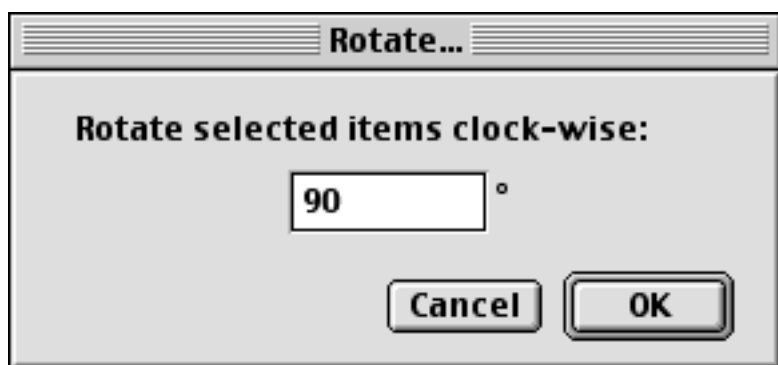


Figure 2-18: Rotate the selected items (but not their icons or text labels) by the specified number of degrees.

Scale: Change the relative spacing of the selected items. This is useful after arranging items in a star to increase or decrease the diameter of the circle. Figure 2-19 shows the interface.

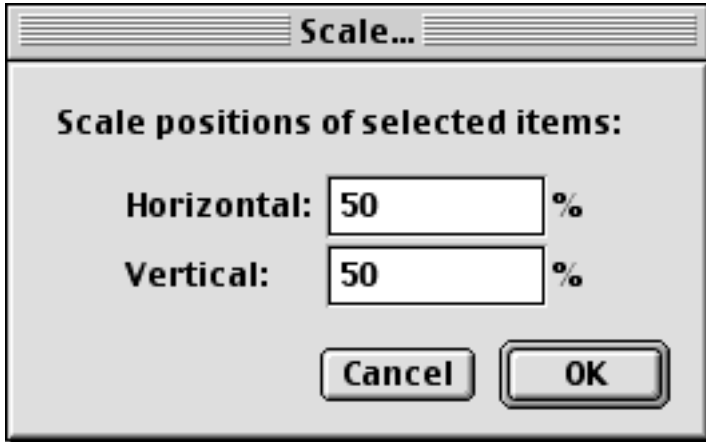
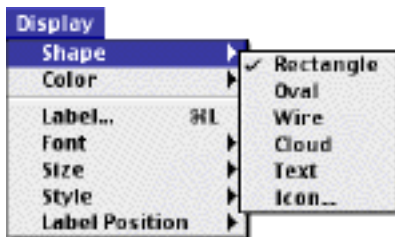


Figure 2-19: Scaling item positions. Specify the amount (percentage) to change the positions both vertically and horizontally.

Display Menu



The **Display** menu contains commands that affect the appearance of individual items in the map. Items can be either devices (routers, servers, hosts, etc.) or networks (drawn as ovals, by default.)

There are commands for changing the shape, color, label, font, and label position.

Shape: Changes an item's shape. The [Choosing a Shape](#) section describes the choices of shape.

Color: Changes the selected items to the desired color.

Label...: (Cmd-L): Modifies the label of one or more items from the map. Devices and networks have text labels that identify the item. These labels may be generated automatically from information gathered from the device, or contain static text that you enter. The [Editing Item Labels](#) section describes how to edit an items' label.

Font, Size, Style: Change the font, size, and style of the items' label.

Label Position: Control the placement of the label relative to an item. There are nine choices: Top Left, Top, Top Right, Left, Center, Right, Bottom Left, Bottom, and Bottom Right.

Choosing A Shape

Devices and networks may take one of several representations on a map.

Rectangle and Oval: These are drawn large enough to contain the text label within them.

Wire: Draw the item as a straight line. Connections to the wire will be drawn at right angles to the wire if possible. Drag the ends of the wire to resize it or change its orientation (angle).

Cloud: Draw the item with a cloud shape. The cloud is drawn to be large enough to contain all the text.

Text: Draw the item as text. The font, style, and color are controlled by the other choices in this menu.

Icon: *InterMapper* has a set of icons which may be applied to devices. Choose **Icon...** to open the window below and select the icon that is desired from the list. See [Appendix D -- Custom Icons](#) to learn more about adding icons to *InterMapper's* set.

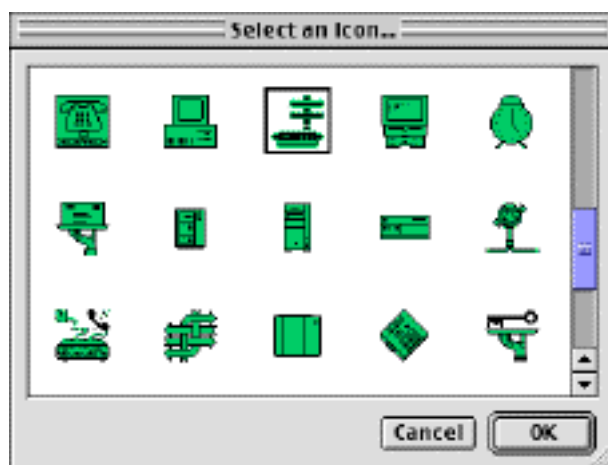


Figure 2-20: Icon Selection Window. Use this window to select an icon for a device or network.

To place a variable into the label, double-click the variable name in the list. It will be inserted into the label

Editing Item Labels

The default label for a device is its full DNS name; the default label for a network is a list of its IP and AppleTalk network numbers. Figure 2-21 below shows the window for editing an item's label.

The top pane lists the label as it will be displayed. The entries in <...> are *variables* which will be filled in with the values from the particular device or network. The lower-left pane displays a list of variables that may be used in the top pane; the lower-right pane shows the definition of each variable.

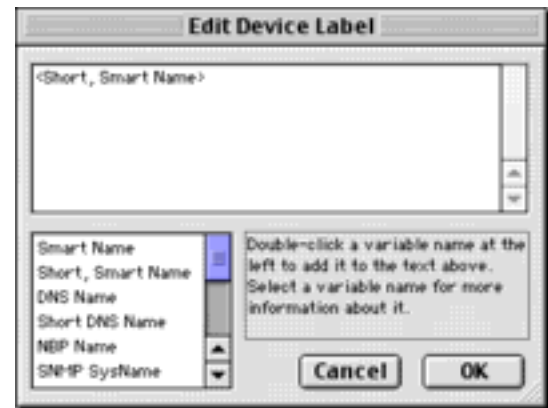


Figure 2-21: Editing a Device label.

text at the cursor position. The following variables are defined for devices:

- Smart Name:** The device's DNS name, SNMP SysName, or address, tried in that order. (Default)
- Short, Smart Name:** The leftmost part (up to the first ".") of the device's Smart Name (except for IP addresses).
- DNS Name:** The full DNS name for the device (not the sysName or IP Address).
- Short DNS Name:** The first part of the device's DNS Name.
- NBP Name:** The device's Name Binding Protocol name.
- SNMP SysName:** The name of the device as reported by the 'sysName' variable.
- SNMP SysDescr:** The hardware and software information reported in the 'sysDescr' variable.
- SNMP SysContact:** The contact person as reported by the 'sysContact' variable.
- SNMP SysLocation:** The location of the device as reported by the 'sysLocation' variable.
- Address:** The network address of the device.
- Probe Type:** The probe type used to test the device.
- Comment:** The comments associated with the device in its "Get Info" window.
- TCP Port:** The TCP port number that is being monitored, if the device is using a TCP-based probe type.

You can also edit the label of a network (oval) using a similar window. The following variables may be used in a network's label:

- Subnet List:** A list of the subnets on the network. (Default)
- IP Subnet List:** A list of IP subnets on the network.
- AT Subnet List:** A list of AppleTalk subnets on the network.
- Port List:** A list of device ports on the network.
- Port Number:** A list of port numbers on the network.
- Port Name:** A list of port names on the network.
- Port Address:** A list of port addresses on the network.
- IP 3rd Octet:** A list of IP subnets on the network, identified by their third octet only.

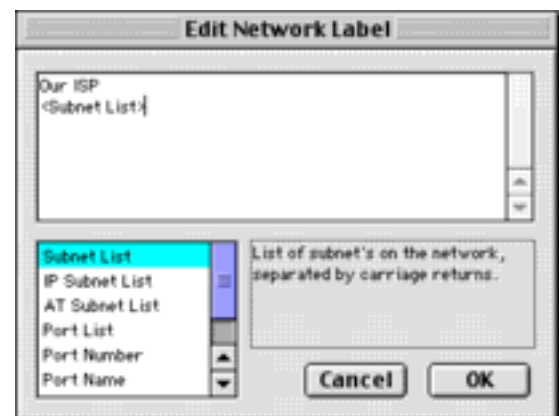
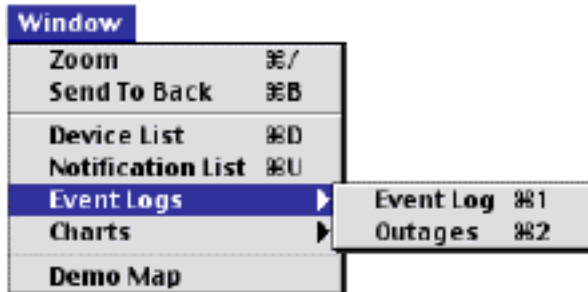


Figure 2-21a: Editing a Network Label

Window Menu



The Window menu lists the open maps at the bottom, and the following commands:

Zoom: (Cmd-/) This expands the frontmost window to the largest size necessary to show all devices, or to the maximum size of its current screen, if all items cannot be visible. Choosing **Zoom** again will return the window to its original size.

Send to Back: (Cmd-B) Sends the front-most window to the back. Floating windows associated with that window will be hidden.

Device List: (Cmd-D) *InterMapper* keeps a list of all the devices it is monitoring. Figure 2-22 shows the status of the devices being monitored in all the open maps.

Status	Name	Condition	Date	Time	Previous Condition	Date	Time
Down	mail.company.net	Down	06/09	11:40:41	Unknown	06/09	11:36:38
Warning	dns.company.net	[DNS] IP address in respons...	06/09	11:47:38	Unknown	06/09	11:47:38
Warning	www.company.net	[HTTP] "<meta>" not found L..	06/09	11:43:43	Unknown	06/09	11:36:38
OK	router.company.net	OK	06/09	11:44:34	OK	06/09	11:36:38
Warning	ftp.company.net	OK	06/09	11:49:05	Not Available	06/09	11:49:05

Figure 2-22: The Device list window.

The columns of the Device List window are:

Status: The device's state. The icon's color matches its color in the map.

Name: The first line of the device's name as shown on the map

Condition: The most severe (i.e. worst) status for the device

Date & Time: Shows when the device entered its current state

Previous condition: The device's status before it entered the current state

Date & Time: Shows when the device entered the previous condition

The default sort order is by device status: click a column heading to sort by that field.

Event Log: (Cmd-E) The **Event Log** windows display a history of alarms, warnings, and up and down and other events. Figure 2-23 shows a typical Event Log window.

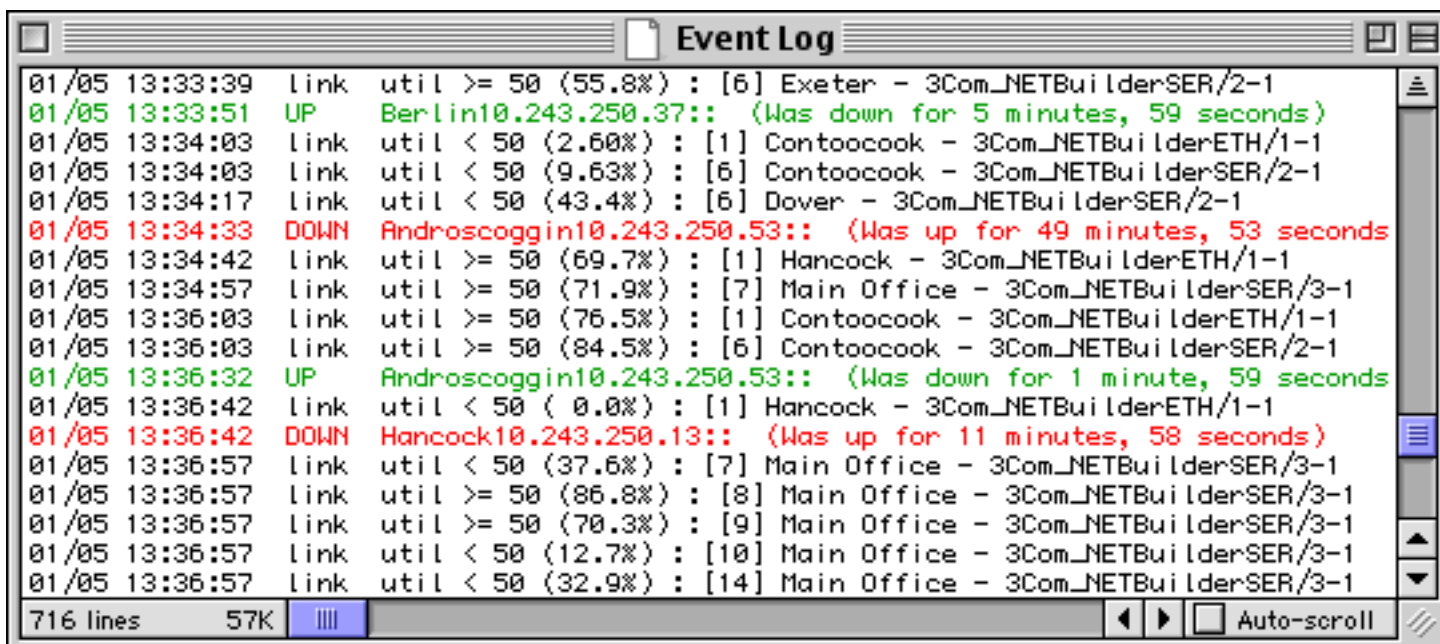


Figure 2-23: Event Log window.

Each time a device changes state, an entry is made in an event log window. In addition, *InterMapper* logs messages for the following events:

- Acknowledgements (including the text entered by the operator)
- Maps opening and closing
- Program startup
- DNS errors
- Errors when sending a notification
- Receipt of an SNMP trap

A further discussion of Event Log windows is on the [Event Log Windows](#) page of Chapter 5.

Notification List: (Cmd-U) This window shows all the methods of notifications that have been defined. This window is described in detail in the [Notification chapter](#).

Charts: This is a hierarchical menu showing all the charts for the current map window. You can select individual charts, or show or hide all charts.



The bottom of the **Windows** menu shows a list of the open maps. The figure at the top of this page shows the "Demo Map" window is open.

Figure 2-23a: The Charts sub-menu.

Notifications

InterMapper can send many different kinds of notifications to alert the network manager of problems in the network. An entire map can be configured for a default notification (or set of notifications), and can then individual devices can have customized notifications.

Each notification has two attributes: the type of notification to send and a schedule of hours during which the notification should be sent. Whenever an event occurs, *InterMapper* scans its list for notifications attached to that device and sends the proper notification to users whose schedule indicates that they should be notified. The window in Figure 3-1 shows the window used to configure the two parts of a notification:

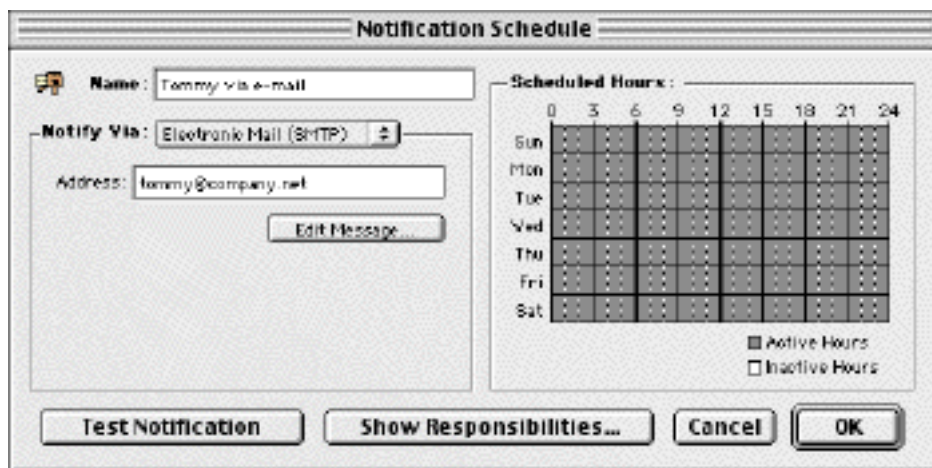


Figure 3-1: A typical notification schedule. This shows an email notification will be sent to "tommy@company.net", at any time -- 24 hours per day and 7 days per week.

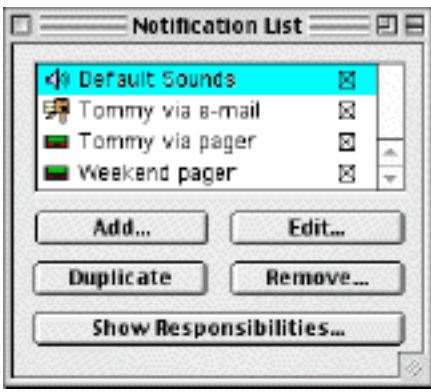
The left side of the window specifies the type of notification. The choices are sounds, e-mail, a radio page, an AppleEvent, or an SNMP Trap. When you select the type of notification from the pop-up menu, the left pane will change to show the required parameters. The right side of the window shows the time schedule for which the notification should be active. To configure a notification:

- Type a name for the notification. This can be any descriptive string. If the notification is active only at certain times of the day or week, you may want to include a description of the time period as well. For example, you could assign names like "Weekend Pager" and "Second Shift Pager" to notifications that had those time schedules.
- Select the type of notification to send from the **Notify via:** pop-up menu.
 - Sound:** Select the sound to be played
 - Electronic mail:** Fill in the recipient's e-mail address
 - Pager:** Fill in the proper pager information
 - AppleEvent:** Select an application to receive the AppleEvent
 - Trap:** Enter the DNS name or IP address of a device that will receive a trap from InterMapper when the device changes state.
- Select a range of hours during which this notification should be sent. Drag across a range of hours to select them. Click individual cells to add or remove hours from the selection. Double-clicking the Active/Inactive Hours legend selects/deselects all hours.
- If desired, edit the message to be sent along with the notification by clicking the Edit Message... button. Figure 3-8 on the [E-mail Notification page](#) shows the editing interface.
- Click the **Test Notification...** button to send a sample notification.

Once you have created notifications schedules, you may attach them to all devices (the default notification is used for all new devices) or one or more devices.

To review the current set of notifications, choose **Notification List** (Cmd-U) from the **Window** menu. Figure 3-2 shows a set of notifications that are active for a map:

The **Show Responsibilities...** button displays a list of the devices that use a particular notification. Figure



Each notification also has a checkbox next to it. Uncheck this box to disable this method of notification. (This is useful for turning off notifications for someone who is going on vacation.) Simply check the box to reactivate the notifications.

To create a new notification, click **Add...** To edit an existing notification, select it, and click **Edit...** Figure 3-1 above shows the window used to specify a notification.

The **Duplicate** button creates a copy of the notification, which can then be further customized. The **Remove...** button deletes the notification.

Figure 3-2: Notification List window. The Default Sounds are built-in.

3-3 show an example:

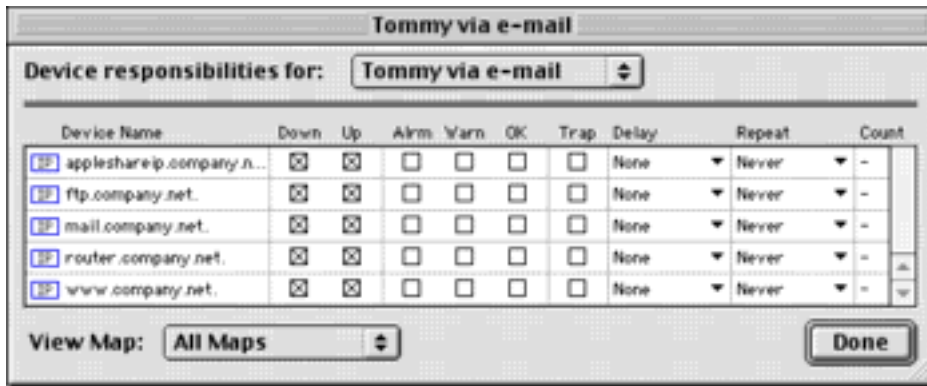


Figure 3-3: Devices that depend on the "Tommy via e-mail" notification.

To set the default notification for newly added devices on the map, choose **Default Notification...** from the **Network** menu.

Setting Notifications for a Device

To view or modify notifications for one or more devices, select the device(s) and choose **Attach Notification** from the **Network** menu. In either event, you will see a window similar to Figure 3-4.

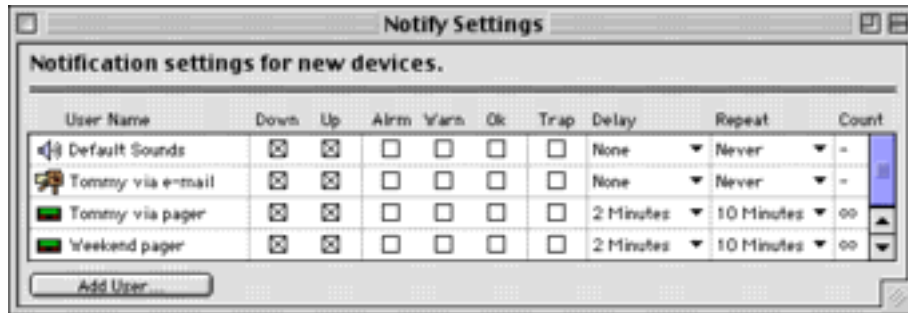


Figure 3-4: Notification Settings for a device.

Check the boxes for each notification that should be sent. The user will receive a notification each time a device changes to the checked state.

You can also modify a particular notification method by clicking the icon to the left of its name. This is equivalent to opening the Notification List (Figure 3-2 above) and clicking the **Edit** button.

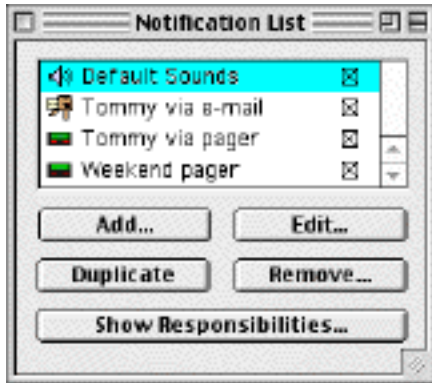
Each notification also has **Delay**, **Repeat**, and **Count** parameters. These can be used to adjust the timing (and frequency) of any notification. For example, the two pagers above are set to wait until the device has been down for two minutes before sending the page, and to retry the page every 10 minutes forever (the count of "-" indicates no ending count). Notifications will be sent until the count is reached, or the device has been acknowledged.

You can achieve problem escalation with notifications by creating two or more notification schedules for a device. The first can fire quickly to notify someone right away. The other notifications can be set to delay for some time, perhaps 30 minutes or an hour before notifying a different person. If the problem remains when the longer delay elapses, the second notification will be sent. If the problem has been acknowledged, no further notifications will be sent, even if the outage lasts a long time.

Audible Alerts

InterMapper can play a sound whenever a device enters a new state. (States are UP, DOWN, WARN, ALARM, OK, and TRAP RECEIVED.) InterMapper's default notification for all devices is to play the *Klaxon* sound when a device goes down, and to play the *Yahoo* sound when it comes back up.

You may control the sound assignments with the Sound Playback schedule. You can alter the default set of sounds, or create custom sound combinations for different kinds of devices.



To do this, open the **Notification Schedules** window from the **Windows** menu. You will see a window like Figure 3-5:

- Double-click the "Default Sounds" item to edit the Default Sounds for every device.
- If you want to create a new set of sounds, click the **Add...** button. You can then assign that new set to various devices.

Figure 3-5: Notifications List.

In either case, you'll see a window similar to Figure 3-6. This shows a pop-up menu for each state that allows you to select the sound to play (or to select "None"), and the volume for that sound.

Mac OS X: You can select any of the system sounds or the sound files in `/System/Library/Sounds` folder.

On the right is the Scheduled Hours grid. Select (by clicking, shift-clicking, and dragging) the hours during which the specified sounds should be active.



Figure 3-6: Schedule for Audible Alerts. A different sound may be assigned to denote the time a device goes into a new state (UP, DOWN, ALARM, etc.)

Notification by E-Mail

InterMapper can send a message to an e-mail message with information about the problem. Figure 3-7 shows a typical E-mail notification window.

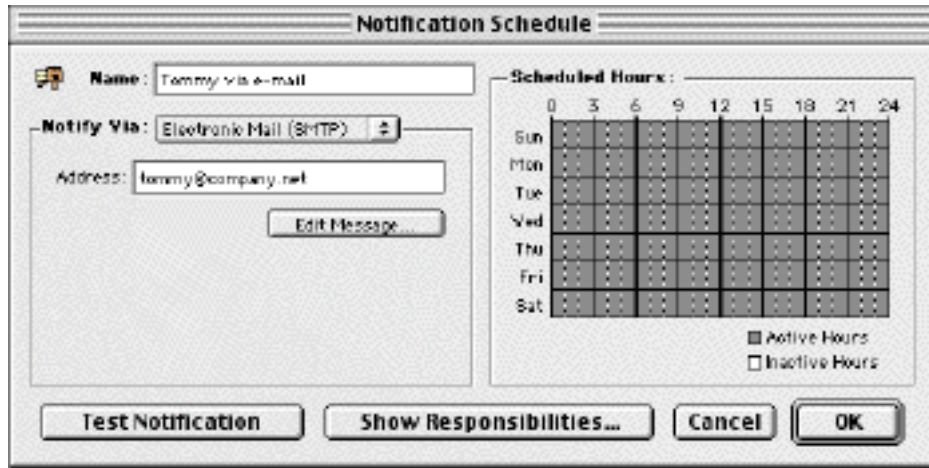


Figure 3-7: Notifications by e-mail.

In the notification Schedule window, select "Electronic mail (SMTP)" from the popup menu, and fill in the e-mail address that should receive the notification.

Note: You must set up a SMTP host to handle the mail in the [E-mail Preferences](#) window.

E-mail and pager notifications carry a text message that describes the failure. For example, the default e-mail notification is shown in Figure 3-8. The list at the lower left contains variables that will be substituted into the text. Double-click an item to have its current value inserted into the text of the message that is sent.



Figure 3-8: Editing the notification text.

Notification by Pager

InterMapper works with the PageNOW! software from Mark/Space Softworks. Here is a brief introduction to installing and using PageNOW! with InterMapper. For more information, visit the Mark/Space website at <http://www.markspace.com>.

First, install the PageNOW! software from floppy disk, CD, or downloaded software. Double-click the PageNOW! application on your hard drive.

Choose the **Subscribers...** command from the **Configure** menu. Figure 3-9 shows a typical window for creating a new subscriber -- the person who will receive the page.



Figure 3-9: The list of PageNOW! subscribers. Each of these entries may be paged from InterMapper or other software running on this Macintosh.

To create a new subscriber, click the **New** button. To edit an existing subscriber, click the **Edit** button. In either case, you will see a window similar to Figure 3-10.

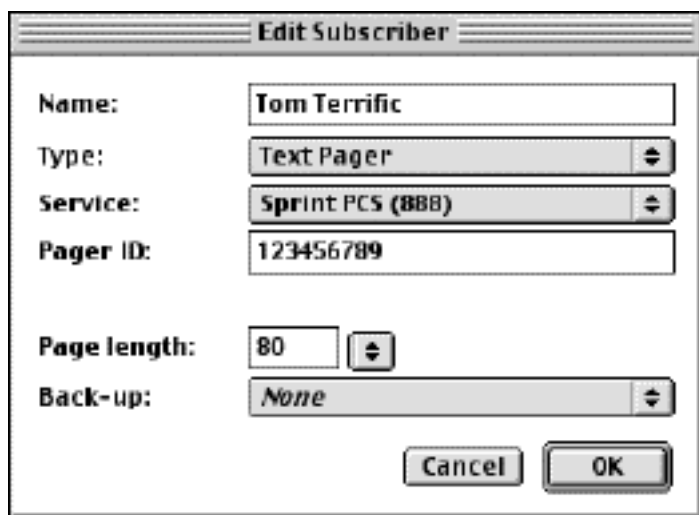


Figure 3-10: Editing a PageNOW! subscriber's information.

Fill in the various fields and pop-up menus. These include:

- Name:** Enter the person's name
- Pager type:** Choose Text or Numeric, Speech, or E-mail
- Service:** Select a pager service from the popup menu. You may choose between traditional analog (modem-based) paging or SNPP or HTTP paging options. If your pager service is not listed, refer to the PageNOW! documentation to learn how to enter information about a new pager service.
- Pager ID:** Enter the pager's ID
- Page length:** Select the number of characters the pager can handle

Click **OK** to retain the information entered.

Open InterMapper, and choose **Notification Schedule** from the **Network** menu. Create a new notification schedule, and select *Alpha-numeric Pager* from the pop-up menu. Figure 3-11 shows a typical schedule.

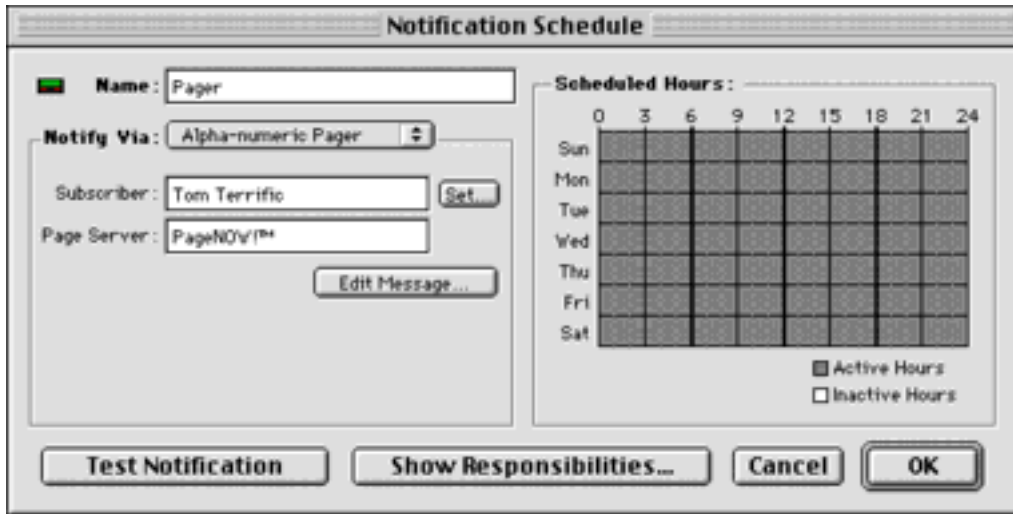


Figure 3-11: A Pager schedule. Enter a name for this individual, select the Subscriber and Page Service by clicking the **Set...** button.

You may type the Subscriber and Page Service entries, or use the **Set...** button to select them from PageNOW's directory. If you click **Set...**, a window similar to Figure 3-12 will open:



Figure 3-12: Selecting a PageNOW! subscriber.

Choose PageNOW! (Local) if the PageNOW! program is running on the same computer as InterMapper. You should see a list of all the subscribers created by PageNOW! Select the desired subscriber, and click **OK**.

Pager message text may also be edited. Click the **Edit message...** button, and make changes as described in the [E-mail notification](#) page.

About PageNOW! PE and Server Editions

The PageNOW! software from Mark/Space Softworks (<http://www.markspace.com>) has two versions. InterMapper can instruct "PE" (Personal Edition) version can page a single individual. The Server edition allows InterMapper to page multiple individuals.

Notifications with AppleScript

InterMapper can run another program on the Macintosh by sending it an AppleEvent. Typically, this is an AppleScript program that has been written specially to handle the message sent from InterMapper. Examples of these AppleScripts are the PageNOW and PowerKey notifiers. This section describes the format of the AppleEvent that is sent, and gives an example of a script that receives the AppleEvent.

InterMapper's events have a class of `iM!r` and a type of `ntfy` (case is important). The `!` and `r` characters are formed by typing Option-\ and Option-Shift-\ on the Macintosh keyboard. You can also copy this example and paste it into the Script Editor to create your own AppleScripts that handle InterMapper's notifications.

When saving an AppleScript for use as a notification, be sure to save it as an Application (classic applet for MacOS 9), and check both the **Stay Open** and **Never show startup screen** boxes as shown in Figure 3-14.

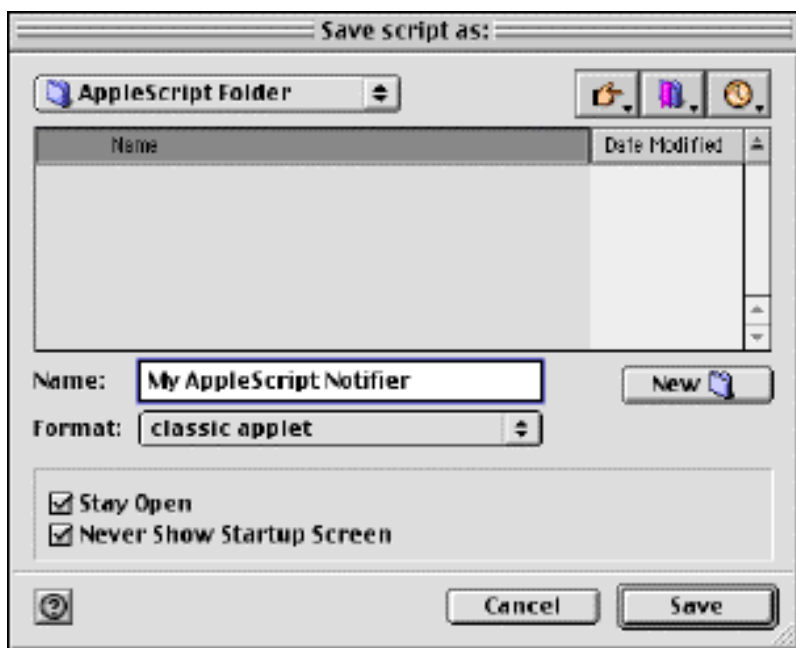


Figure 3-14: Save as... application, with the **Stay Open** and **Never show startup screen** boxes checked. (MacOS 9.0 uses "classic applet" in place of "application".)

InterMapper passes a number of parameters in the AppleEvent. These are described in the sample AppleScript below, as well as the example script that is saved in the *AppleScript Info* folder in the *InterMapper Settings* folder.

```
(*
This script demonstrates how InterMapper passes notification information to a script or
application.
```

```
For this script to work most efficiently, you should save it as an 'Application' with the
'Stay Open' and 'Never show start-up screen' checkboxes set.
```

```
InterMapper sends an AppleEvent with class 'iM!r' and type 'ntfy' including the
following parameters (all passed as strings):
```

```

user_name - (keyword = '----') - the name of the user who requested notification.
event_type - (keyword = 'Evt') - the type of event. Possible values are 'DOWN',
            'UP', 'Alarm', 'Warning', and 'OK'
device_name - (keyword = 'Name') - the name of the device.
device_addr - (keyword = 'Addr') - the device's IP address or AppleTalk address
```

```
probe_type - (keyword = 'Prob') - the device's probe type (e.g. SNMP, ICMP Echo)
device_condition - (keyword = 'Cond') - the device's condition string
device_sysUpTime - (keyword = 'Uptm') - how long the device claims to have been running
                    (basedon sysUpTime).
timestamp - (keyword = 'Time') - the time when InterMapper sent the event.
last_down - (keyword = 'Down') - the time the device was previously down
```

InterMapper does not expect a reply from the 'iM!r' AppleEvent.

The "on run" handler calls the Notification handler with the parameters you type in the parameter list. It is useful for debugging the handler before you try it out with InterMapper.

*)

```
-- This handler allows you to debug the script from the Script Editor simply by
-- running it (with Cmd-R) Or check out Smile, a free AppleScript editor and
-- development environment (http://www.tandb.com.au/smile/)
```

```
on run
```

```
set result to («event iM!rntfy» of "Notifier's name in Notification List"
    given «class Evnt>:"DOWN", «class Name>:"www.foo.com", «class Addr>:"127.0.0.1",
    «class Prob>:"HTTP", «class Cond>:"Reason for outage",
    «class Uptm>:"dd days, hh hours, mm minutes, ss seconds",
    «class Time>:"mm/dd hh:mm:ss", «class Down>:"MM/DD HH:MM:SS")
```

```
result -- display the result of calling InterMapper's Notification routine
```

```
end run
```

```
on «event iM!rntfy» user_name given «class Evnt>:event_type, «class Name>:device_name,
    «class Addr>:device_addr, «class Prob>:probe_type, «class Cond>:device_condition,
    «class Uptm>:device_sysUpTime, «class Time>:timestamp, «class Down>:last_down
```

```
-- InterMapper passes the following parameters:
```

```
--
```

```
-- user_name = name of the user being notified
-- event_type = type of event taking place
-- device_name = name of the device
-- device_addr = IP or AppleTalk address of the device
-- probe_type = how the device is being polled
-- device_condition = the device's condition (for ALRM and WARN events)
-- device_sysUpTime = how long the device claims to have been running
-- timestamp = when the notification was sent
-- last_down = time the device was last down
```

```
display dialog timestamp & ": " & event_type & " - " & device_name
```

```
end «event iM!rntfy»
```

Corrective Action with PowerKey Pro

InterMapper can send commands to a PowerKey Pro to turn power on and off to a particular outlet (thus, restarting a server that may have failed.)

InterMapper ships with an AppleScript that can receive notifications regarding a device, and turn the proper outlet off and then on. The script leaves the outlet off for approximately 10 seconds to be sure that all devices on the computer have been turned off before restoring the power. This script is in the InterMapper's *AppleScript Info* folder.

When it sends a notification, InterMapper passes the IP address of the device as a parameter to an AppleScript. The 'outlets' array below maps an IP address to a PowerKey Unit and Outlet number. The entries are, in left-to-right order:

```
{ "IP-Adrs-of-device", "PowerKey-Unit-Number", "PowerKey-Outlet-Number" }
```

You are responsible for editing this script to match your own setup. To customize the script for your own PowerKey, enter your own IP addresses, unit numbers and outlet numbers. If you do not need to control all five outlets, you may remove the extra lines. Or add more lines if you have several PowerKeys to control by supplying a different PowerKey-Unit-Number.

```
set outlets to { ↵
  {"192.168.1.10", "1", "2"}, ↵
  {"192.168.1.11", "1", "3"}, ↵
  {"192.168.1.12", "1", "4"}, ↵
  {"192.168.1.13", "1", "5"}, ↵
  {"192.168.1.14", "1", "6"} ↵
}
```

Notification by Traps

InterMapper will send a SNMP Trap as a notification when a device goes into a particular state.

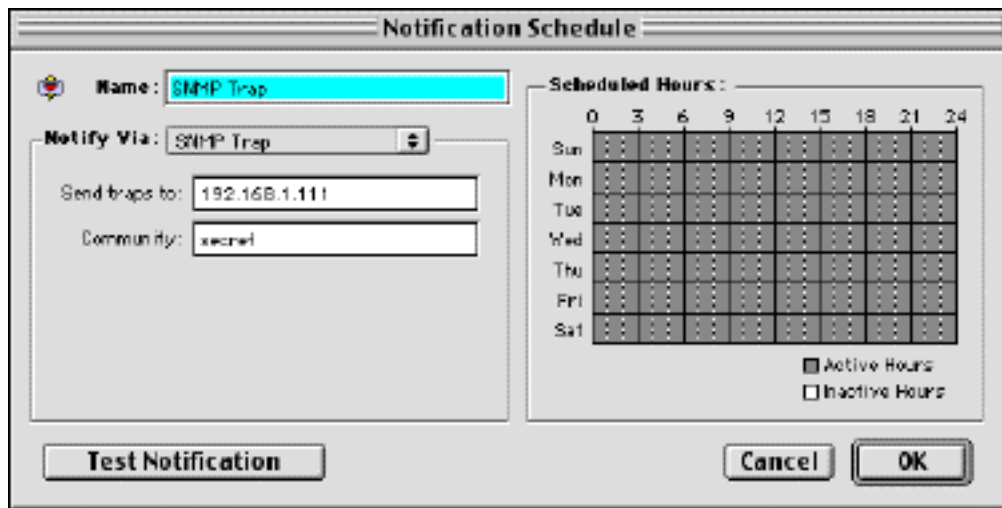


Figure 3-13: Trap Notification Schedule. Enter the IP Address or DNS name for the device to receive the trap, and the SNMP Trap Community String.

In the notification Schedule window, select "SNMP Trap" from the popup menu, and fill in the IP address or DNS name of the device to receive the trap, and the SNMP Trap Community string.

InterMapper sends four pieces of information in the trap:

- Timestamp:** The current date and time, as a string
- Message:** DOWN, UP, ALARM, WARN, OK, or TRAP
- Device name:** The devices DNS name, as a string
- Condition:** The condition of the device, as it would be printed in the log file.

InterMapper's traps contain the following MIB variables:

```
intermapperTimestamp = 1.3.6.1.4.1.6306.2.1.1.0
intermapperMessage = 1.3.6.1.4.1.6306.2.1.2.0
intermapperDeviceName = 1.3.6.1.4.1.6306.2.1.3.0
intermapperCondition = 1.3.6.1.4.1.6306.2.1.4.0
```

All are encoded as OCTET STRING. This information is also available in [ASN.1 format](#).

The Dartware MIB

Dartware LLC has registered the Enterprise 6306 for its own SNMP variables. The remainder of this page shows the Dartware MIB in ASN.1 notation.

```
DARTWARE-MIB DEFINITIONS ::= BEGIN

IMPORTS
    enterprises
        FROM RFC1155-SMI
    DisplayString
        FROM RFC1213-MIB
    OBJECT-TYPE
        FROM RFC-1212
    TRAP-TYPE
        FROM RFC-1215;

-- Define SNMPv1 Traps sent by InterMapper 3.0 and later
-- (23 October 2000)

dartware          OBJECT IDENTIFIER ::= { enterprises 6306 }
notify            OBJECT IDENTIFIER ::= { dartware 2 }
intermapper       OBJECT IDENTIFIER ::= { notify 1 }

intermapperTimestamp OBJECT-TYPE
    SYNTAX          DisplayString (SIZE(0..255))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "The current date and time, as a string."
    ::= { intermapper 1 }

intermapperMessage OBJECT-TYPE
    SYNTAX          DisplayString (SIZE(0..255))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "The type of event: DOWN, UP, ALARM, WARN, OK, or TRAP."
    ::= { intermapper 2 }

intermapperDeviceName OBJECT-TYPE
    SYNTAX          DisplayString (SIZE(0..255))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "The device's DNS name, as a string."
    ::= { intermapper 3 }

intermapperCondition OBJECT-TYPE
    SYNTAX          DisplayString (SIZE(0..255))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "The condition of the device, as it would be printed in the log file."
    ::= { intermapper 4 }

intermapperTrap TRAP-TYPE
```

```
ENTERPRISE      dartware
VARIABLES      { intermapperTimestamp, intermapperMessage, intermapperDeviceName,
intermapperCondition }
DESCRIPTION
    "The SNMP trap that is generated by InterMapper as a notification option."
 ::= 1
```

END

Notifications based on Receiving Traps

InterMapper can also act as a SNMP trap receiver. It records information about all the received traps in the log file. InterMapper can also filter the traps to determine when notifications should be sent, and to whom, based on the information contained in the trap. To do this, InterMapper passes the trap to an AppleScript which analyzes the trap information, and then determines how to send the trap.

There are four steps required to process a device's trap by an AppleScript (you may want to keep InterMapper open as you follow along):

1. First, create an AppleScript that will process the SNMP Traps as they arrive. You'll need to create a script to handle the particular kinds of traps your devices will send. The [SNMP Trap Filter](#) could provide a basis for your script. The comments in the script give more information about how traps are processed or filtered. Be sure to save the script as a stay-open application/applet, with no startup screen.
2. Next, create a new notification to be used with these traps. To do this, choose Notification List from the Windows menu. Click Add... You will see a Notification Schedule window.
3. Give a name (e.g., use "Traps") to this notification and set it to notify via "AppleEvent/AppleScript". Click the Set... button, and select the "SNMP Trap Filter" file or the AppleScript that you modified. You'll probably want to leave this notification active all the time. This notification is now available to be used by any device on a map.
4. Finally, attach this trap notification script to a device. To do this, select (click on) the device and choose **Attach Notification...** from the Network menu. A "settings" window will open, showing all the notifications that have been defined (and in particular, the "Traps" notification created above.)

Sample Trap Filtering Script

```

-- This example script handles traps arriving from a device and delays sending a notification
-- unless two minutes have passed and the device has not returned to normal operation.
--
-- This script illustrates three techniques:
--
-- o How an AppleScript can process a SNMP trap
-- o How a notification AppleScript can delay for a time period
--   before initiating an action
-- o How an AppleScript can trigger an InterMapper notification

-- 7 March 2001 reb

property ALARM_USER : "Tom Terrific" -- name of user to notify (selected from the Notification List window)
property ALARM_DELAY : 120 -- duration in seconds to wait before sending alarm
property ALARM_BEGIN_STRING : "(1234)" -- substring in condition indicating beginning of alarm
property ALARM_CLEAR_STRING : "(4321)" -- substring in condition indicating alarm is cleared

property gAlarmString : "" -- non-empty string is used to store alarm condition
property gAlarmTime : 0 -- stores time we entered the alarm condition

-- This handler is invoked when InterMapper calls this AppleScript as a Notification.
-- InterMapper sends the «event iM!rntfy» AppleEvent with the parameters below.

on «event iM!rntfy» user_name given «class Evt»:event_type, «class Name»:device_name, «class Addr»:device_addr, «class Prob»:probe_type, «class Cond»:device_condition, «class Uptm»:device_sysUpTime, «class Time»:timestamp

-- InterMapper passes the following parameters:
--
-- user_name = name of the user being notified (from the Notification List)
-- event_type = type of event taking place ("TRAP")
-- device_name = name of the device on the map
-- device_addr = IP or AppleTalk address of the device
-- probe_type = how the device is being polled (may not be relevant)
-- device_condition = the device's condition string (see below)
-- device_sysUpTime = how long the device claims to have been running (may not be set if not polled with
SNMP)
-- timestamp = time the notification was sent
--
-- InterMapper passes in a device_condition string that contains two parts:
--
-- The OID of the trap, followed by the specific trap type in parentheses
-- A list of other variables that were sent in the trap packet separated by ",",
-- and enclosed in curly braces (" {...}")
--
-- For example, an APC uninterruptible power supplies send traps containing two items:
-- the trap type and a single "mtrapargsString" text string that describes the trap
-- InterMapper would pass this trap information in the condition string as this text:
--
-- 1.3.6.1.4.1.318 (5) UPS: Switched to battery backup power; utility power failure.

if (event_type is not "TRAP") then
-- We only handle "TRAP" events.
return
end if

-- Set gAlarmString to a non-empty error message to signal the alarm condition. Remember the time
-- of this event. The script's idle handler periodically checks if there's an alarm condition
-- that hasn't been cleared within the 120 seconds.

-- Check the device_condition to see if its trap type matches the ALARM_BEGIN_STRING. If so, enter the alarm
state
-- Else check to see if it contains the ALARM_CLEAR_STRING, and if so, clear the alarm state (it didn't time
out)
-- If it contains neither, then pass the trap information back to InterMapper for normal notification
processing

if device_condition contains ALARM_BEGIN_STRING then

```

```

-- Enter Alarm.
set gAlarmString to timestamp & " -- " & device_name & " : " & device_condition
set gAlarmTime to the current date

else if device_condition contains ALARM_CLEAR_STRING and gAlarmString is not "" then
-- Clear Alarm.
set gAlarmString to ""

else
-- Pass all other TRAPs through.
set theAlarmString to timestamp & " -- " & device_name & " : " & device_condition
-- tell application "InterMapper[trademark] 3.0.4-PPC" to
---
try
    tell application "Finder"
        set appPath to application file id "iM!r" as string
        set appName to name of application file id "iM!r"
    end tell

    launch application appPath --does not open untitled window

    tell application appPath
        tell application appName
            activate
            -- notify user_name Message theAlarmString Subject theAlarmString
        end tell
    end tell

    on error number -1728
        error "Could not find InterMapper application."
    end try

    ---
end if

end «event iM!rntfy»

-- This idle handler runs every 10 seconds to see if the gAlarmString is non-empty,
-- and if 120 seconds have elapsed since the gAlarmTime was set.
-- If so, then it's time to send off the alarm.

on idle

-- Check the alarm string. If it hasn't been cleared
if (gAlarmString is not "") then
    set currttime to time of (current date) -- 'time' is number of seconds since midnight.
    set lasttime to time of (gAlarmTime)

    if (currttime - lasttime > ALARM_DELAY) then
        -- Alarm wasn't cleared in time, so send delayed alarm notification.
        tell application "InterMapper[trademark] 3.0.4-PPC"
            Notify ALARM_USER Message gAlarmString Subject gAlarmString
        end tell
        -- Clear Alarm.
        set gAlarmString to ""
    end if
end if

return 10 -- go to sleep for 10 seconds
end idle

-- This handler allows you to debug the script from the Script Editor simply by Running it (with Cmd-R)
-- Or check out Smile, a free AppleScript editor and development environment (http://www.tandb.com.au/smile/)

on run

set result to («event iM!rntfy» of "Notifier's name in Notification List" given «class Evnt>:"DOWN", ¬
    «class Name>:"www.foo.com", «class Addr>:"127.0.0.1", «class Prob>:"HTTP", ¬
    «class Cond>:"Reason for outage", «class Uptm>:"dd days, hh hours, mm minutes, ss seconds", ¬
    «class Time>:"mm/dd hh:mm:ss", «class Down>:"MM/DD HH:MM:SS")

result -- display the result of calling InterMapper's Notification routine

end run

```

InterMapper Preferences

InterMapper uses a single window for its categories of preferences. To select a category, use the **Category:** pop-up menu at the top of the window shown in Figure 4-1.

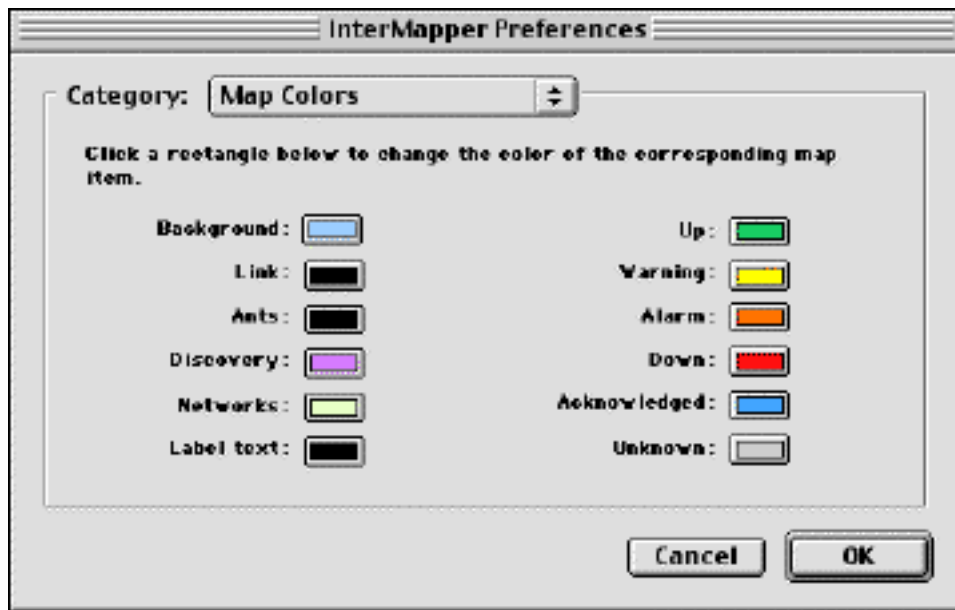


Figure 4-1: *InterMapper Preferences Window*. The **Category** popup menu at the top allows you to select which type of preferences will be changed. Selecting an item in the **Category:** popup menu changes the information shown in the pane below it.

InterMapper has preferences for setting:

- [Default map colors](#)
- [SNMP Read-only Community String](#)
- [Log file preferences](#)
- [DNS servers to use for resolving names](#)
- [E-mail information for sending e-mail notifications](#)
- [Global web pages and the telnet server](#) access lists
- [Miscellaneous preferences](#)
- Default appearance of [devices and networks](#) and [strip charts](#)
- [Users and Groups](#) that may access InterMapper's servers.
- [Debugging preferences](#)
- [Colors for an individual map](#)
- [Web access controls for an individual map](#)

Default Map Colors Preference

When InterMapper creates a new map, it uses various colors for the items and features on the map.

This window allows you to change a color for any feature on a map. To do this, click in that feature's box. The standard Macintosh Color Picker will open and allow you to choose the desired color. Changing the default colors will not change the colors assigned to an existing map.

Tip: You may change an individual map's color assignments with the [Map Colors](#) preference.

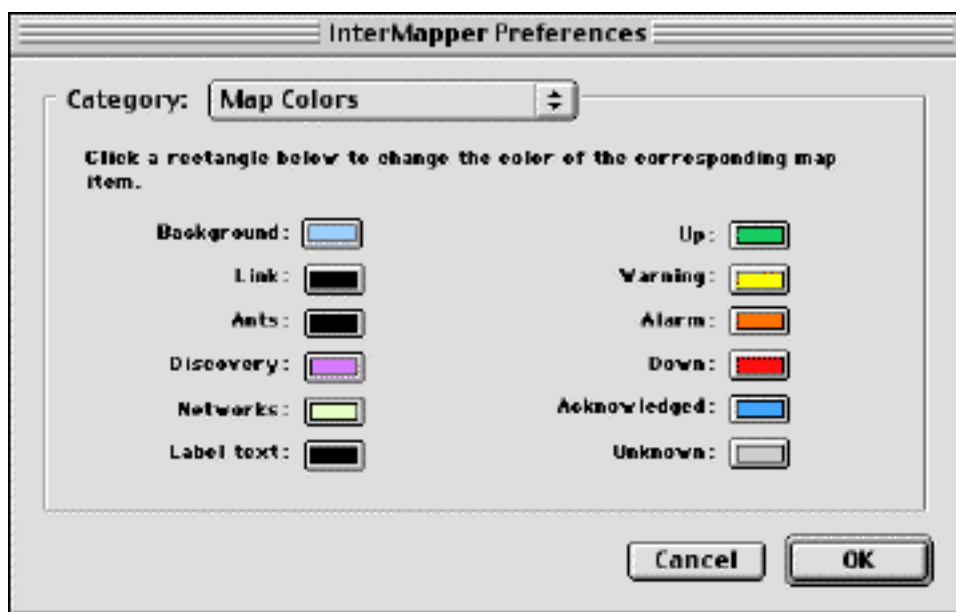


Figure 4-3: The Color Preference window. Click on any of the colors to open the Color Picker and select a different color for that device/link.

SNMP Community String

The SNMP Read-only Community string is like a user id or password that allows access to a router's or other device's statistics. InterMapper sends the community string along with all SNMP requests. If the community string is correct, the device responds with the requested information. If the community string is incorrect, the device simply discards the request and does not respond.

There are actually three community strings for SNMP-speaking devices:

- The SNMP Read-only community string enables a remote device to retrieve "read-only" information from a device. InterMapper uses this to request information from devices on its maps.
- The SNMP Read-Write community string allows a remote device to read information from a device and to modify settings on that device. InterMapper does not use the read-write community string, since it never attempts to modify any settings on its devices.
- The SNMP Trap community string is used when sending SNMP Traps to another device. InterMapper accepts any SNMP Trap community string.

By convention, most equipment ships from the factory with a read-only community string of "public". It is standard practice for network managers to change all the community strings so that outsiders cannot see information about the internal network. (In addition, network managers may employ firewalls to block any SNMP traffic to ports 161 and 162 on the internal network.)

Figure 4-2 shows the preference window for changing the default SNMP community string, which is initially "public". Changing this value causes InterMapper to use the new string when querying SNMP devices.

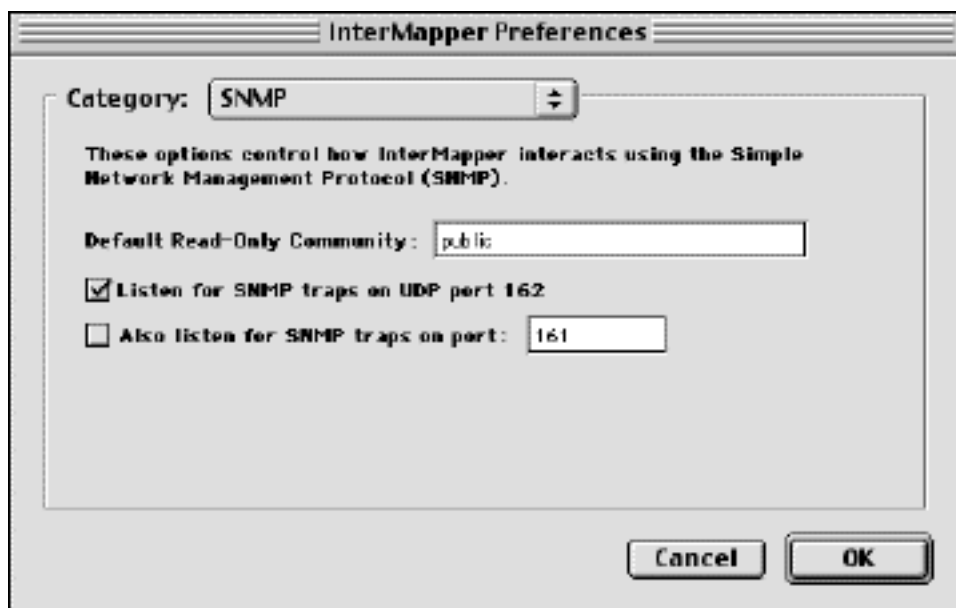


Figure 4-2: Setting the default SNMP Community string.

Individual devices on a map may have different SNMP Community strings. See the [Device Information Window](#) page for setting an individual device's read-only community string, or the [Set Info...](#) command for setting multiple devices at one time.

InterMapper can also listen for traps on a second, non-standard port (in addition to port 162). Check the box and enter the port number in the field indicated. Traps received on this alternate port will be handled in the same manner as those received on port 162.

Log File Preferences

InterMapper keeps log files of events that occur. These text files are saved in the *InterMapper Settings:InterMapper Logs* folder. Each file has a user-defined name that describes its function, and ends with a suffix of *.yyyymmddhhmm*, where the suffix is the (four-digit) year, month, day, hour and minute of the file was created.

Within InterMapper, there are several sources of logging information. These sources include up and down entries for the devices being logged, hits on the built-in web server, connections to the InterMapper Remote and Telnet server, InterMapper's own internal status and error messages and others.

There are three built-in log files that are always present, and cannot be deleted.

- The **Error Log** file. When InterMapper first executes, this file receives all entries from all sources. The network administrator can divert certain streams to other log files.
- The **Outages** file. This contains entries that describe the start and end time of outages, as well as their duration. This stream of entries cannot be redirected to any other log file.
- The **Debug** file. This displays certain debugging information, which is described in the [Debug window](#) page.

These streams of log entries can be sent a particular log file. By default, all entries go to the built-in Error Log file. Many log file entries (such as web server hits, or Telnet server connection messages) may be redirected to other log files. This is done using the appropriate server configuration preference pane.

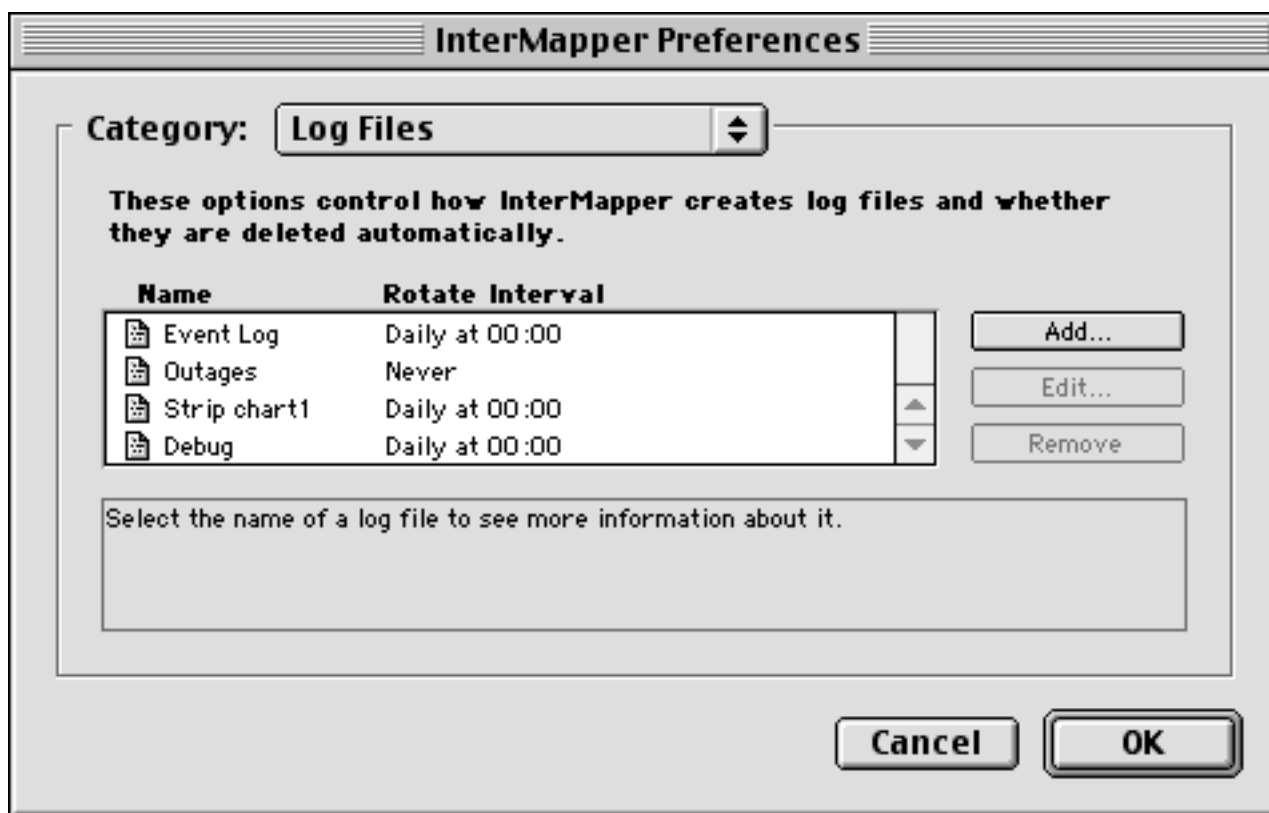


Figure 4-4: Logging Preferences.

Figure 4-4 above shows a typical log file preferences window. It shows the names of the log files, and their rotation intervals.

To add a new log file, click the **Add...** button. To edit information about a log file, select the log file name, then click **Edit...** In either case, you'll see a window similar to Figure 4-4a.

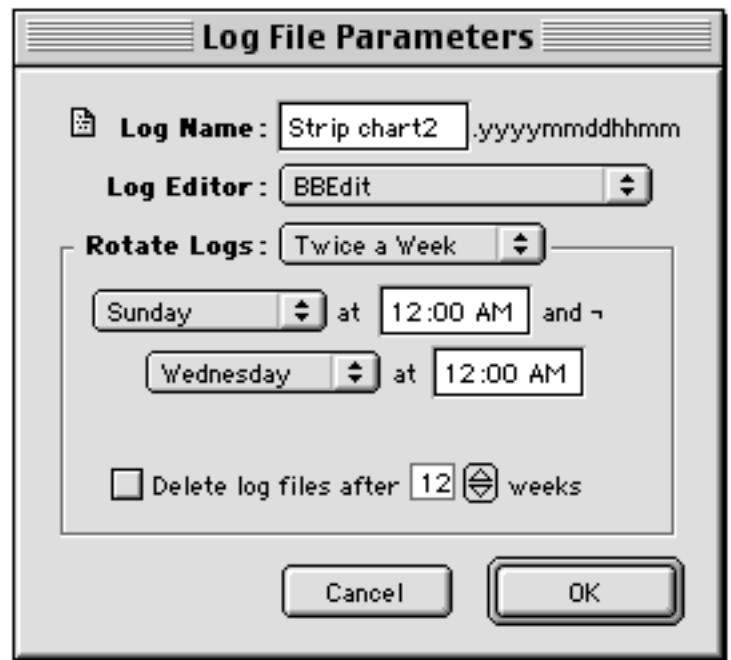
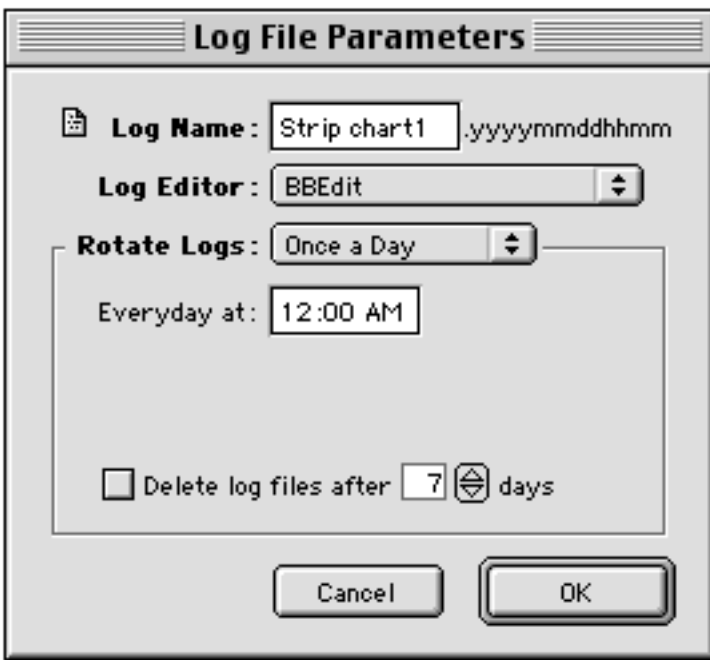


Figure 4-4a: Log file preferences. Two different examples of log file parameters.

The figure on the left shows the parameters for "Strip Chart 1". It has these options:

Log Name: Set the name for the log file. It may be up to 18 characters long (because there are 13 characters in the `.yyyyymmddhhmm`, and file names may only be 31 characters long. **Log Editor:** The file will be saved with a type of 'TEXT' and a creator that allow it to be opened by the BBEdit text editor.

Rotate Logs: File rotation consists of closing the current log file and opening a new at some time to break the log files into convenient sizes and/or time epochs. Choices in this popup menu are "Never", once or twice daily, and once or twice weekly. When file rotation is specified, you must select a time to rotate the file as well. The right image of Figure 4-4a shows a log file that will be rotated twice each week.

Delete log files after ___ days/weeks Check this box to force InterMapper to delete old log files automatically after a certain date. C

DNS Monitor Preferences

InterMapper requires a reliable Domain Name Service server (DNS) for its operation. InterMapper uses the DNS server(s) to check that the DNS name and IP address for a device match at regular intervals.

For example, when *InterMapper* polls devices that have a name assigned, it looks up the corresponding IP address in the DNS. If this address has changed since the device was added to a map, InterMapper will log an error message.

This window allows you to specify the DNS server(s) that InterMapper will use. InterMapper default is not to use a DNS server. The servers used by MacTCP or TCP/IP are never used. Figure 4-5 shows the settings for configuring the DNS monitor.

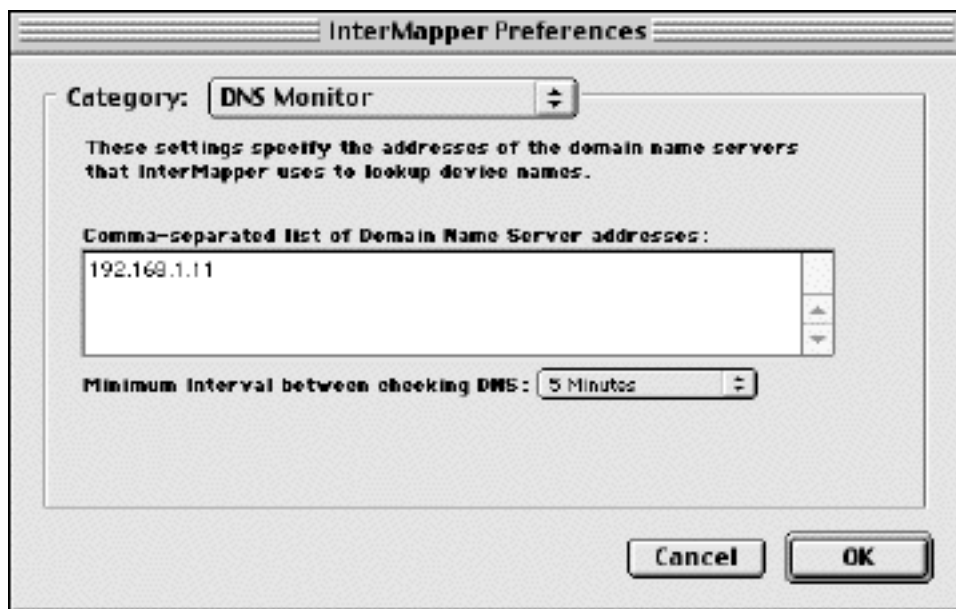


Figure 4-5: DNS Monitor preferences. Specify the DNS server to use for name queries, and an interval to check names.

The **Minimum interval between DNS checks** field allows you to set the time to wait between successive queries for a host. To minimize the rate at which domain names are checked, you can set this popup menu to a larger value.

E-Mail Preferences

Use this category to enter the information *InterMapper* requires to send e-mail notifications.



The screenshot shows a dialog box titled "InterMapper Preferences". At the top, there is a "Category:" dropdown menu set to "E-Mail". Below this, a text box explains: "These options control how InterMapper sends notifications using e-mail. The SMTP host specifies the address of your mail server." There are two input fields: "Primary SMTP Host:" with the value "mail.service1.com" and "Back-up SMTP Host:" with the value "mailer.otherserver.com". Another text box explains: "The Sender's address is the e-mail address that will appear in the From: field of messages sent by InterMapper." Below this are two more input fields: "Sender's Address:" with the value "my-name@company.com" and "Errors to:" with the value "my-name@company.com". At the bottom right, there are "Cancel" and "OK" buttons.

Figure 4-6: Configuration for sending e-mail notifications.

SMTP Hosts: Enter the name of one or two hosts that will accept (and possibly forward) e-mail notifications. InterMapper will try the Primary host first, and if unsuccessful, will attempt to deliver e-mail messages through the Backup host.

Sender Address: Enter the e-mail address of the person/account who should be shown as the sender of the e-mail notification. This address will be listed in the "Sender:" field of the message.

Errors to: Enter the e-mail address of the person who should be notified about problems delivering e-mail notifications. This is the name listed in the "Errors-to:" field of the message header.

Web and Telnet Server Preferences

InterMapper can act both as a Web server and a Telnet server to make information about the network available via these services. Both the Web and Telnet servers use similar windows. Figure 4-7 shows the Telnet server configuration window; the Web configuration window is similar. In addition, you may set up access controls for individuals or groups as described Figures 4-8 and 4-9.

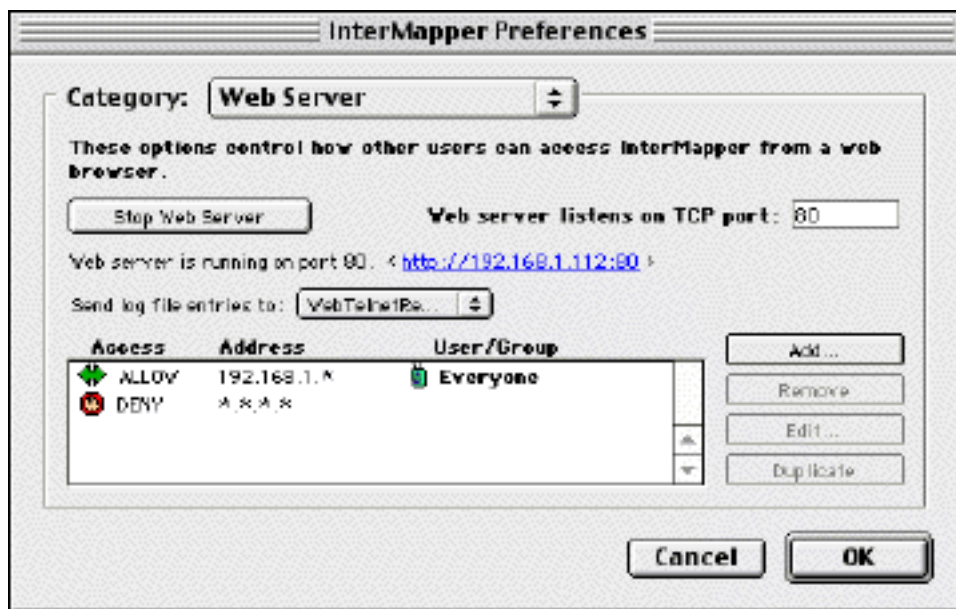


Figure 4-7: Web Server configuration (the Telnet server window is similar). Click the Start button to enable web (telnet) service on the specified port. You should also create one or more entries in the access list.

Start/Stop button. Click this button to start (stop) the server. When you start the server, the status line contains a blue link to the current server. Clicking it will launch a web browser/telnet application to connect to InterMapper.

Telnet (Web) server listens on port ____. Enter the port that the built-in Web (Telnet) server will use to accept connections. Default HTTP port is 80; the default Telnet port is 23.

Send log file entries to.... Select a log file to receive the entries generated by the web server.

Access list: A list of address ranges and individuals that will be allowed or denied access to the Web or Telnet server. *InterMapper* checks the list in top-to-bottom order. If the address or person matches, and the access is "Allow", access is granted; if the access is "Deny", then it is denied. If the end of the list is reached without a match, access is denied.

Addresses in the access control list may have three different formats:

- Fully-specified IP addresses, such as 192.168.1.10
- Address ranges, such as 192.168.1.1-31. This would allow any device in the range 192.168.1.1 to 192.168.1.31 to connect.
- Addresses with an "*" as a wildcard that corresponds to a range of 0-255. This allows addresses of the form 192.168.1.* (equivalent to 192.168.1.1-255), or a Class B range of (192.168.*.*), or the "all addresses" range of *.*.*.*

In addition, InterMapper can require a name and password from the web browser or telnet user before allowing access. Add users either to the MacOS Users and Groups file (MacOS Classic, and computers that are not running as a AppleShare IP server) or InterMapper's own built-in preferences. You can select which to use in the [Users and Groups Preferences](#).

Reloading Web Files

The InterMapper web server reads its web template and target files at two times:

- Whenever InterMapper starts up
- Whenever the **Web Server** web server starts up.

When you're editing these files (as described in [Appendix C -- Customizing Web Pages](#)), we recommend that you leave the Preferences window open. After you've made a change, simply click the **Stop** button to shut down the web server, then click **Start**. This takes a couple seconds. InterMapper will reload the web files and they become available immediately.

Adding Entries to the Access List

To add a new entry to the Access List, click the **Add...** button. To edit an entry, click the **Edit...** button. In either event, a window like Figure 4-8 or 4-9 will open:

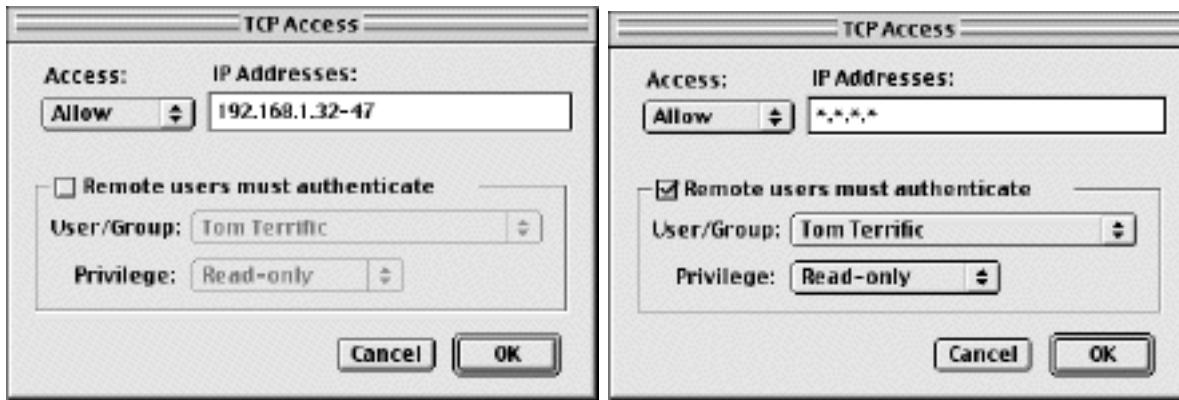


Figure 4-8 and 4-9: Setting up the Access List. The left window shows that anyone from the IP address range 192.168.1.32 to 192.168.1.47 may connect. The right window shows that Tom Terrific can connect from any address, but only after entering his name and password.

You can combine IP address ranges as well as authentication access controls to narrow the set of people/machines that may access the server as shown above.

Note: Name/password authentication for telnet connections is not implemented.

Tip: You may change an individual map's access list with the [Map Settings](#) preference.

InterMapper Remote Preferences

InterMapper can act as a server to support the [InterMapper Remote](#) client. InterMapper Remote gives you a real-time view on your network from a remote computer. You can see devices going up and down, the live ants, etc. (This differs from the Web facility, since the images on the web pages are static.)

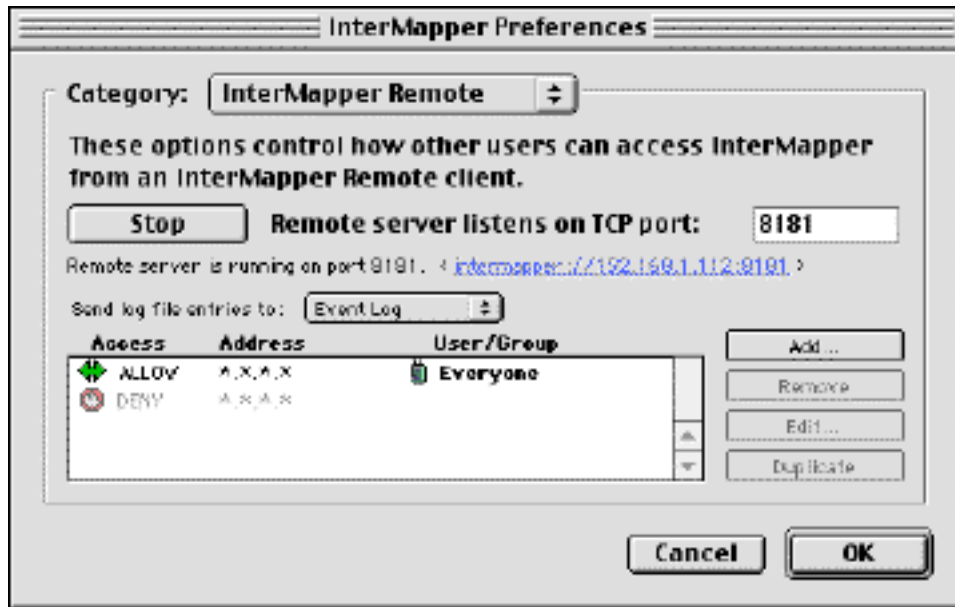


Figure 4-9a: InterMapper Remote server configuration.

The InterMapper Remote server configuration works just like the [Web and Telnet servers](#). As with these other servers, you can configure one access list for the entire machine, and then create separate access permissions for individual maps using the [Map Settings](#) preference.

Note: You cannot use the built-in MacOS Users and Groups as authentication for InterMapper Remote. Instead, use the InterMapper "Remote Users" facility

The InterMapper Remote server is available as part of InterMapper 3.6 and newer.

Miscellaneous Preferences

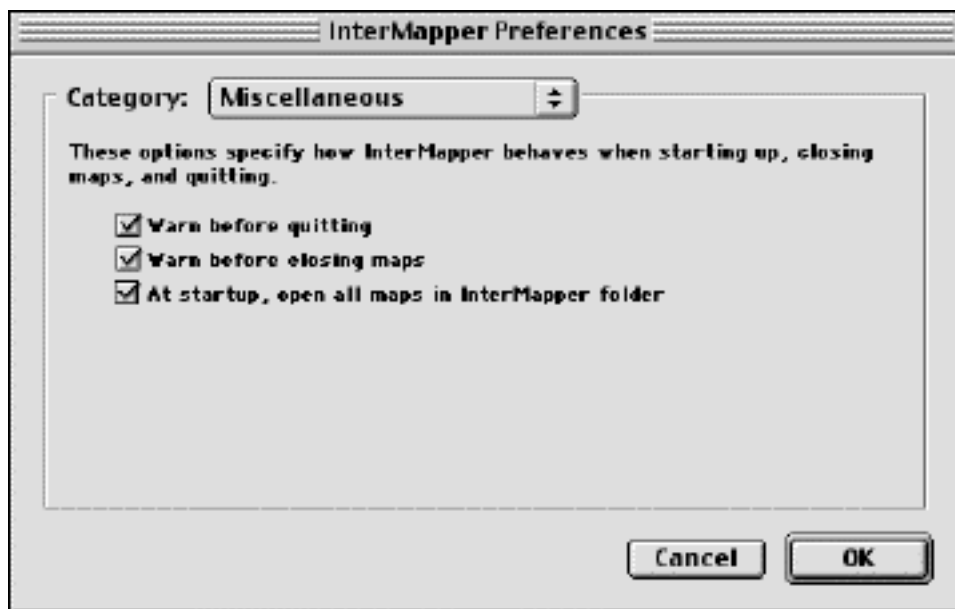


Figure 4-10: Miscellaneous Preferences.

By default, *InterMapper* will warn you before quitting to avoid inadvertent loss of data. Uncheck the **Warn before quitting** box to disable that warning.

Similarly, *InterMapper* will warn you before closing a map. Uncheck the **Warn before closing maps** box to disable that warning.

Finally, *InterMapper* will open all the maps in its folder when it starts up. This is convenient for opening the same set of maps each time *InterMapper* starts. Uncheck the **At startup, open all maps in InterMapper folder** box to disable this behavior.

Default Device and Network Preferences

When devices are first added to the map, *InterMapper* will display them as rectangles, by default. *InterMapper* also defaults to showing networks as ovals. Figure 4-12 shows the window which changes the default device appearance. A similar dialog is used for setting default network appearance.

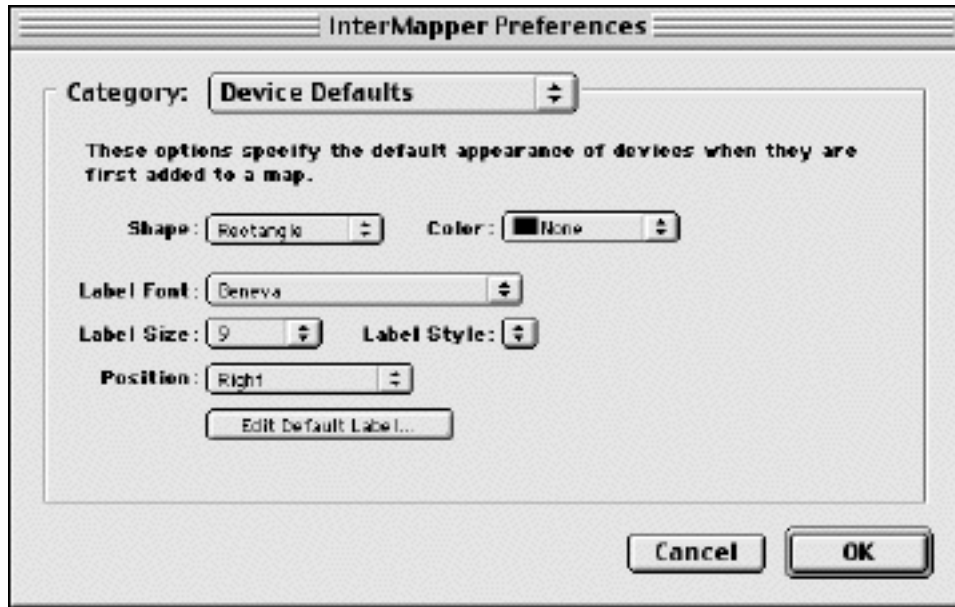


Figure 4-12: Default appearance for devices. Use these settings to change the way devices appear in the map when they are first added to the map. The Default Network window is similar.

Shape Choose one of the shapes shown in the Network menu's Shape choices.

Color Choose a color from this pop-up menu.

Default Label Characteristics Choose a default font, size, style and label position.

Edit Label... Edit the default label, using the facility described in [Chapter 1 -- Modifying Device and Network Labels](#)

Strip Chart Preferences

Strip charts (see [Chapter 1 -- Creating Strip Charts](#)) can show historical data for MIB variables from one or more devices. Use the settings shown in Figure 4-13 to modify the default settings for a newly-created strip chart.

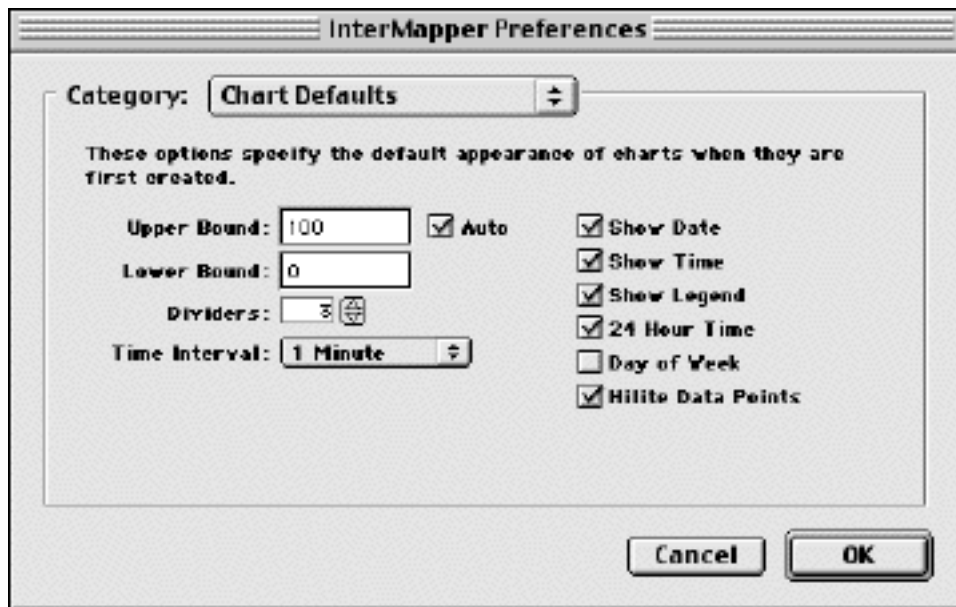


Figure 4-13: Default Chart options.

Upper and Lower Bounds. These control the vertical scale of the chart. Check the *Auto* button to have *InterMapper* adjust the scale automatically. If checked, *InterMapper* will change the upper and/or lower bounds so that data points will always be displayed, no matter how much they increase or decrease.

Dividers. Strip charts have horizontal dividing lines to help interpret the data. Use this to change the number of dividing lines shown on the strip chart.

Time Interval: This is the interval between time stamps on the X-axis of new strip charts. Shorter intervals will show finer detail, longer intervals show a longer history. Because *InterMapper* 3.6 now saves all the data points, so there is no limit on the amount of memory needed to save a strip chart. There is no benefit in choosing a longer time interval - all the data points will be saved.

Show Date, Time, Legend, 24 Hour Time, Day of Week. These affect the labels shown on the strip chart.

Hilite Data Points. If checked, *InterMapper* will draw boxes around the endpoints of the strip chart's line segments.

Remote Users Preferences

InterMapper has a built-in authentication database that controls access to its servers. The organization is similar to the MacOS Users & Groups facility, but is implemented independently.

Users are identified by a name and password. When a connection for a protected resource arrives, the server requests the user's name and password. If these are confirmed in the authentication database, then access is allowed.

Groups are a collection of users. Certain maps may allow anyone in a group to access a resource: the authentication first determines which user is connecting in (using the name and password), and then determines whether that individual is present in the group.



Figure 4-13a: Users and Groups may use the Classic MacOS or InterMapper's own Users and Groups list

This window allows you to add and remove users to the authentication database, and to collect users into groups.

Creating Users and Groups

To create a user or group, click the **New User...** or **New Group...** button. You will see a window like one of these shown below.

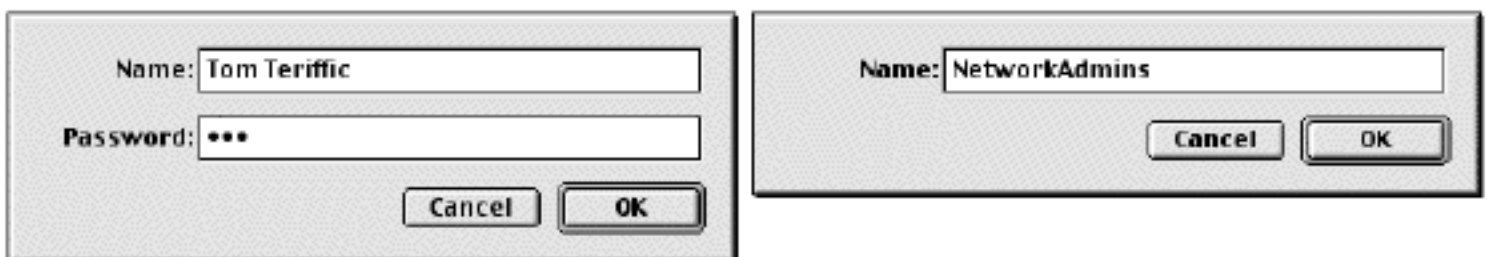


Figure 4-13b: New User and New Group dialogs.

When creating a new user, you must supply the password that goes with that user name. Groups simply have names, without passwords. Click **OK** and you will see the new user/group appear in the window. Users have a single "face" icon; groups have two faces.

Adding and Removing Users to Groups

To add a user to a group, simply drag the icon of the user to the desired Group's icon. Click the disclosure triangle next to a group's icon to view all the users in that group.

To remove a name from the access list or from a group, select the name, and click the **Remove...** button.

To change a user or group name, or a user's password, select the name and click **Edit...**

Using the MacOS Users & Groups Database

Earlier versions of InterMapper allowed access to the Web and Telnet servers based on the built-in Macintosh OS *Users & Groups* database. This worked well for those two kinds of servers, but could not be used for the InterMapper Remote client. (*Technical background:* The InterMapper Remote authentication protocols requires access to the clear-text password; the MacOS Users & Groups does not provide this capability.)

We recommend that everyone employ InterMapper's own built-in database. For backward compatibility, you may check the **Use MacOS Users & Groups for Web and Telnet access** box; this will force InterMapper to use the MacOS database for these servers.

Debugging Preferences

InterMapper has built-in bug detecting and reporting capabilities. *InterMapper* has many internal consistency tests in which it checks that its assumptions are correct. If they are not, it will display a yellow window like Figure 4-11a.

There are four choices for recovering from the error:

- **Log message, Send Email, & Continue:** *InterMapper* will save debugging information to a file on the desktop, then send it as e-mail to the developers at Dartware, and then continue operation.
- **Log message & Continue:** *InterMapper* will save debugging information to a file on the desktop and then continue operation.
- **Continue:** *InterMapper* will not save any debugging information, but will still continue operation.
- **Force-Quit:** *InterMapper* will immediately quit, as if you had typed Command-Shift-Option-Escape. All unsaved changes to maps will be lost.

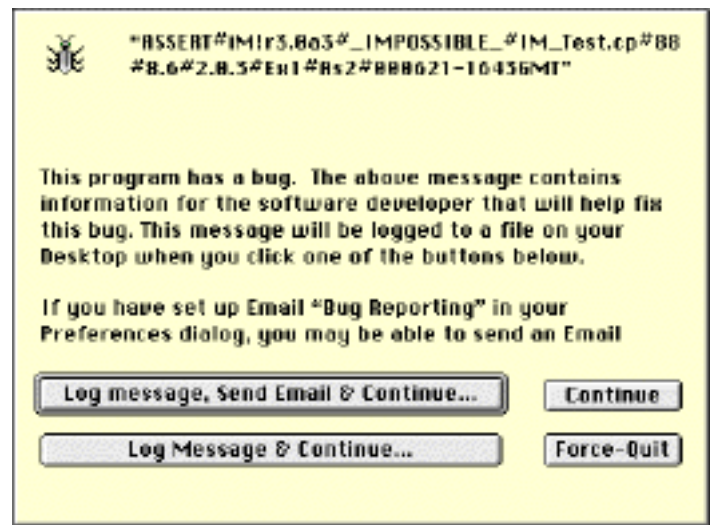


Figure 4-11a: The Debugging Assertion window. This window appears when *InterMapper* discovers an internal error.

Using any of the first three choices is usually safe. Use the **Force-Quit** option if using any of the other choices repeatedly causes another debug window to occur.

We love to have you send this information to us: it helps us find and fix bugs so that they don't afflict other customers. If you wish to install MacsBug, you can retrieve it from the Apple server at <http://developer.apple.com/tools/debuggers/MacsBug/>

Configuring the bug detection and reporting

If MacsBug (a Classic Macintosh system debugging utility) is installed, *InterMapper* can be configured to save certain information which can be helpful in debugging problems. If MacsBug is not present, *InterMapper* will attempt to recover from the program error ("bug"), or exit cleanly, but will not report the problem.

InterMapper has three settings which can affect the amount of debugging information that will be saved. Figure 4-11 shows these options:

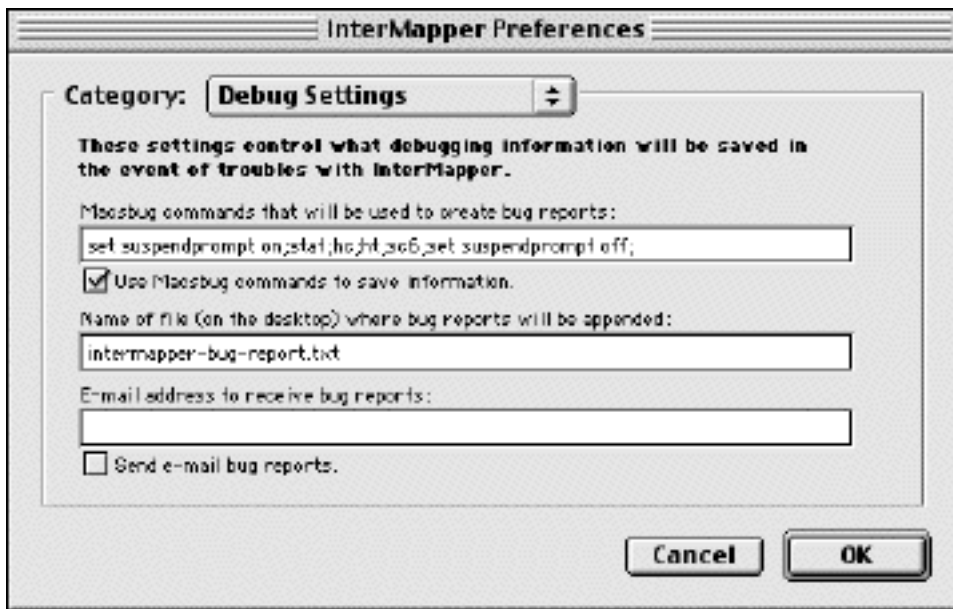


Figure 4-11: Debugging Preferences.

- MacsBug commands that will be used to create bug reports:** The default string supplied provides the usual debugging information. Dartware personnel may ask you to include other commands in this string if the default commands don't display the required information. For reference, the default string is:

```
set suspendprompt on;stat;hc;ht;sc6;set suspendprompt off;
```

Note: MacsBug is only available with Classic MacOS. The other options below are available for both Classic and MacOS X.

- Name of the file (on the desktop) where bug reports will be appended:** InterMapper will save (or append) debugging information to a text file on the desktop. Enter the file's name here. The default file name is:

```
intermapper-bug-report.txt
```

- E-mail address to receive bug reports:** If a mail host has been specified in the [e-mail preference](#) section, InterMapper will send the debugging information directly to Dartware if the **Send e-mail reports** box is checked. The default e-mail address is: intermapper@dartware.com

Note: Be sure that the E-mail settings are correct in the [E-mail Preferences](#) window.

Reporting Crashes in MacOS X 10.1 and later

With MacOS X 10.1 and later, you can configure the Console application to save a log file that will contain useful information about the problem.

This file will be saved in the ~/Library/Logs folder.

To enable this option, Open the Console application (in the /Applications/Utilities folder) and select the Preferences. Check both boxes as shown in this image. Information about subsequent crashes will be saved in files named *crash.log*. There are details about this file later on this page.

You can mail the resulting file to support@dartware.com.

Please include a short description of what was happening when the failure occurred, and an explanation of anything unusual that you think might help us.

Reporting Crashes in MacOS X 10.0.4 and earlier

With MacOS X 10.0.4 and earlier, please turn on the CRASHREPORTER feature of MacOS X. To do this, use the TextEdit application or vi from the Terminal application to open the "/etc/hostconfig" file.

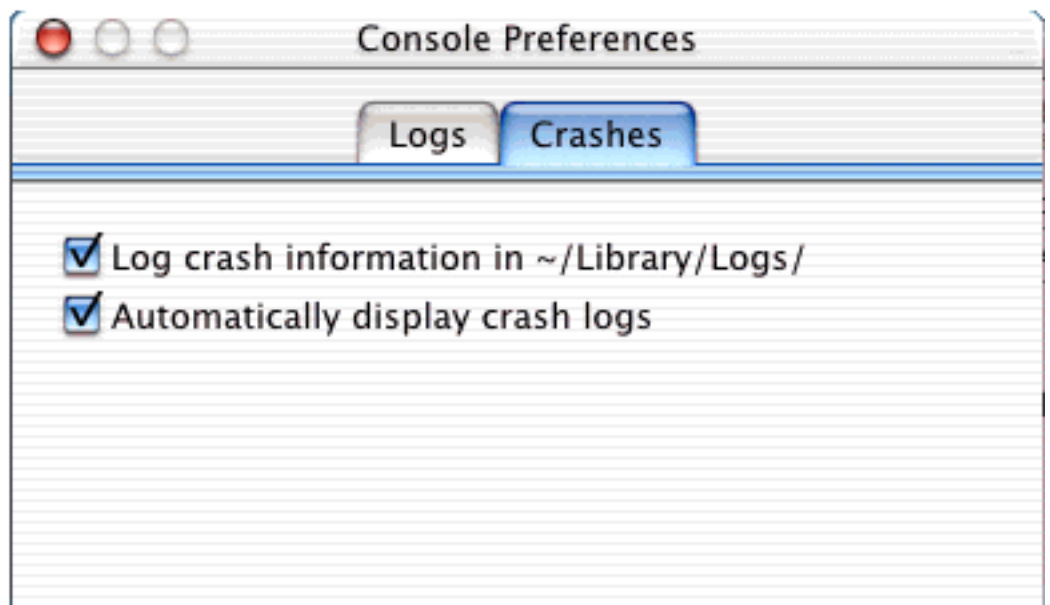
(You must be root or an Administrator to change this file.) Add the following line to the file, save it and restart MacOS X:

```
CRASHREPORTER=-YES-
```

MacOS X - the crash.log file

The CRASHREPORTER feature displays a dialog and writes the pertinent details of a crash to your "/var/log/crash.log" file. The "crash.log" file can be viewed with TextEdit, the Console application or in the Terminal window.

The /var folder is hidden by default in MacOS X. To view it in TextEdit, choose the **Go To Folder...** from the Finder's **Go** menu. Type in "/var/log" in the ensuing window, and you'll see the files there. Drag the "crash.log" file to the TextEdit application, and it'll open.



Map Color Preferences

InterMapper has a global color scheme that is controlled by the [default color preference](#) window. This color scheme applies to all new maps.

Individual maps may have different color settings. These are set by the **Map Settings...** command of the **Edit** menu. It will change the color scheme of the top map (only).

Choose **Map Colors** from the **Category:** popup menu as shown in Figure 4-14. If you want the colors of this map to be different from the global color scheme, check the **Use Map-Specific Colors** box, and choose the desired colors as described in the [default color preference](#) window.

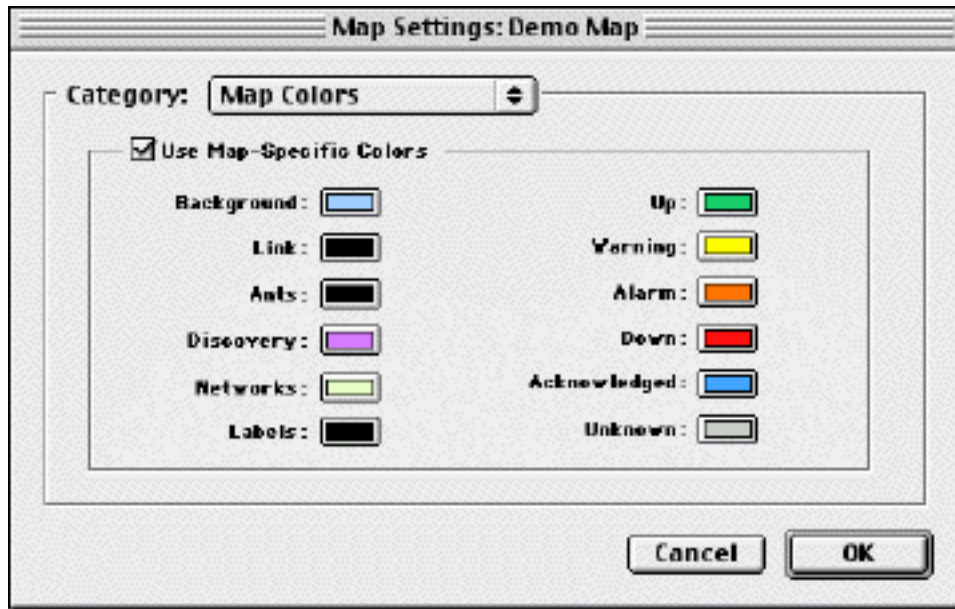


Figure 4-14: The Map Color pane. Check the "Use Map-Specific Colors" box and click any of the boxes to change that color for this map (only).

Controlling Access to Individual Maps

In addition to the [global access list](#) which allows or denies access to all the maps served by the InterMapper program, the network administrator can create a separate access list for each individual map. This could allow certain individuals (who do not have global access) to see certain maps, yet deny access to other maps. (This would be useful for an ISP, who wants a customer to see their own map, not other customers' maps or the global maps.)

The **Map Settings...** command in the **Edit** menu controls this. The **Web Server** choice in the **Category:** popup menu displays the access list, that works like the global web access list.

Controlling Web Access

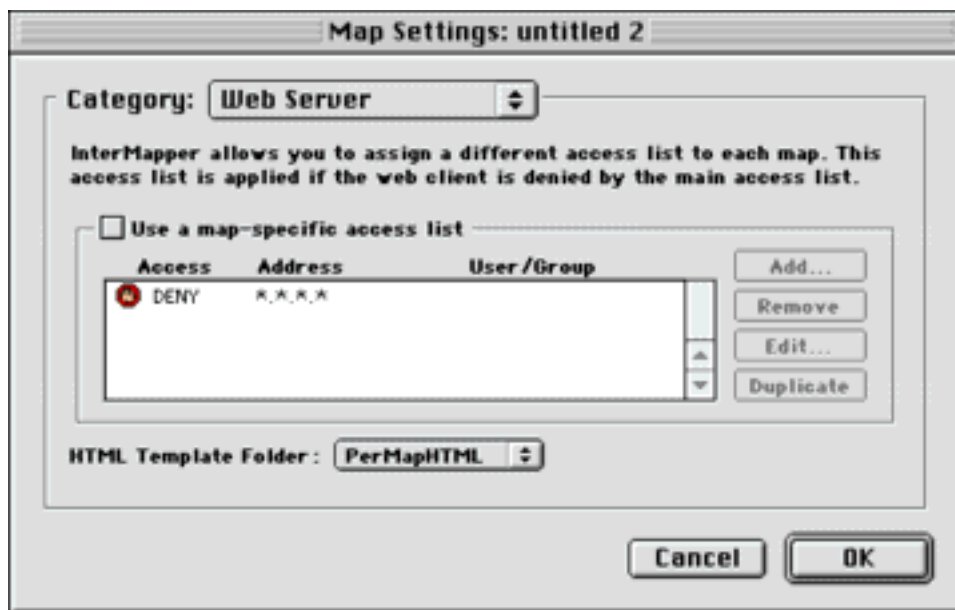


Figure 4-15: Web Server for an individual map. Create a list of individuals who may access this particular map. This figure shows that only Fred may access the map, although he may do it from any IP address.

The **Template Folder:** popup menu selects which set of templates will be used to display the map.

Controlling InterMapper Remote Access

The InterMapper Remote server can control who accesses each map using an InterMapper Remote client program. Use the **Remote Server** category of the **Map Settings...** preference to set up access to each map.

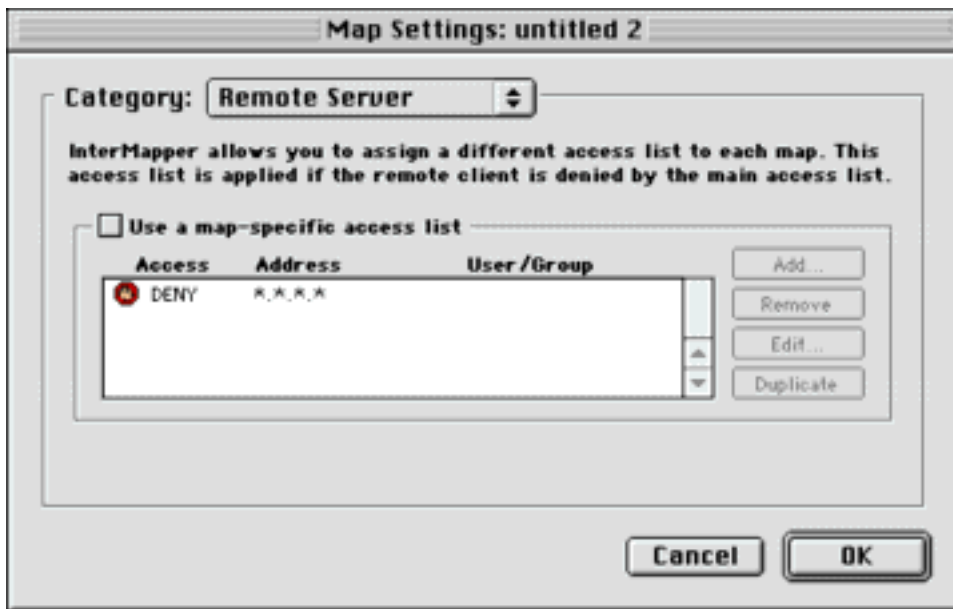


Figure 4-15a: Remote Server for an individual map. Create a list of individuals who may access this particular map. This figure shows that only Fred may access the map, although he may do it from any IP address.

Check the **Use map-specific access list** box to enforce the list of permissions shown. If that box is not checked, InterMapper uses the global **InterMapper Remote** settings in the main **Preferences** pane.

Background Images

InterMapper 3.6 brings the ability to place a background image on a map so that it appears behind the map contents - the devices, icons, and links on the map. These images might show the locations of equipment in an office or against a map of a city or country. The two figures below show a map before and after placing an image in the background.



Figure 4-16a: Map without background image. Figure 4-16b: Same map with background image.

To place an image in a map, choose **Map Settings...** from the **Edit** menu. Select **Background** from the Category popup menu. You will see a window similar to Figure 4-17.

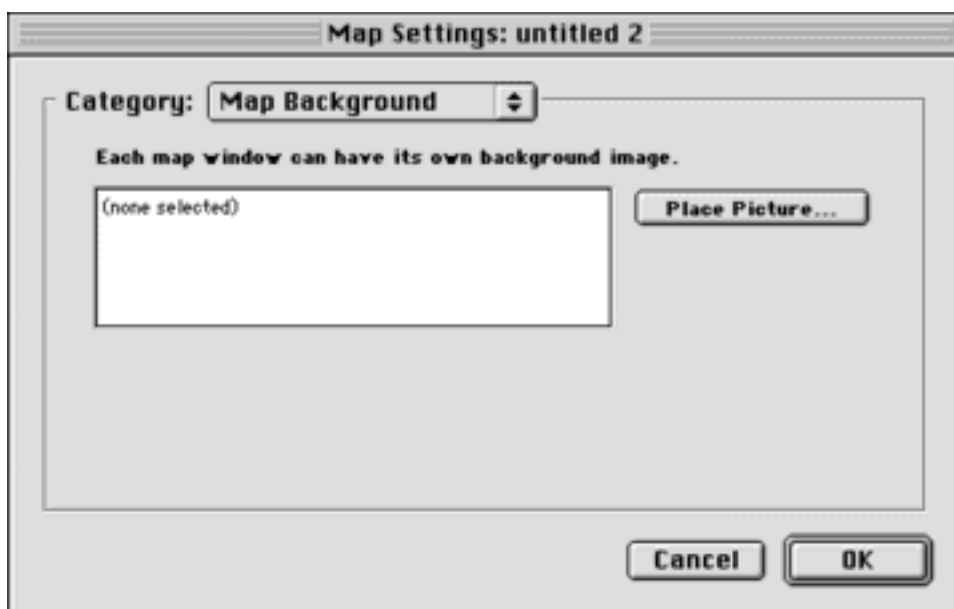


Figure 4-17: Placing an image in a window.

Click **Place Picture...** and select a file. Any image file that can be converted by QuickTime may be used as an image.

The background image becomes a backdrop against which the map contents can be arranged. Once placed in a certain position on the background image, an item will not move unless dragged to a new location.

The background image retains its height and width, and will not be scaled (stretched or shrunk) when you resize the window. If the background image is smaller than the current window size, the image will be centered in the map, and the map's background color will show around the edges. If a large image is placed, its dimensions determine the full size of the window.

Tip #1: Contrasty images (such as Figure 4-16b above) may make it difficult to see the devices and links against the background. To make the image more suitable as a background image, you may use a graphics program to increase the brightness and/or decrease its contrast before placing it in a map. We regularly use GraphicConverter, an inexpensive shareware graphics program from <http://www.lemkesoft.com>, to do this task. It has a Brightness/Contrast adjustment facility to simplify this task.

The figures below show the result of progressively decreasing the contrast and increasing the brightness in the background image before placing it in the map. Notice how it's easier to see the devices and links on the final figure at the right.

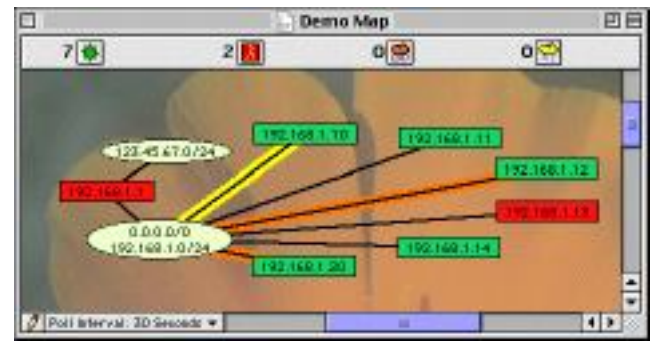
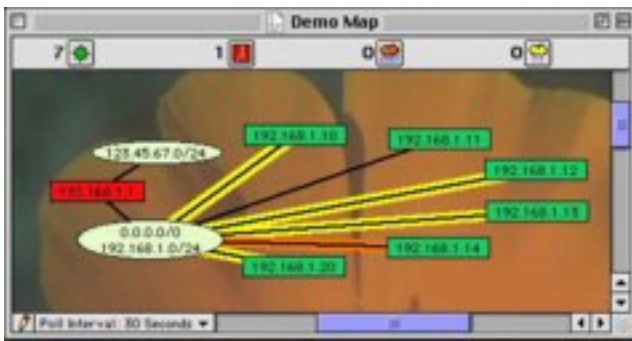


Figure 4-18a: Progressively lighter background image. Figure 4-18b: Lightest background image.

Tip #2: Large images consume large amounts of memory and slow InterMapper's redrawing of the window. You should balance the image quality against the size of the map. Larger maps may look better, however they may consume large amounts of memory. (But see the next tip below.)

Tip #3: Decreasing contrast can also decrease the size of an image, so that decreasing the contrast as described in Tip #1 above may help decrease the size of the background image as well.

Information Windows

InterMapper displays information about devices and about networks in windows that can be opened to give details, or left closed to prevent clutter.

The [Device Information](#) window describes an item on the map. The window includes static information, such as the device's names and network address, its status, etc. and also allows the network manager to change the specifics of how InterMapper tests the device, and how it is represented on the map.

The [Network Information](#) window describes a *network* - the interconnection between devices. The window shows which subnets are associated with a network on the map. It also allows the network manager to add or modify the network's information and appearance, and to scan the network for new devices.

[Event Log Windows](#) show a history of the events such as UPs, DOWNS, Alarms, Warnings, errors, accesses to the Web, Telnet, and InterMapper Remote servers, internal errors, and others.

The [Outages Window](#) shows a history of the outages that InterMapper detects.

[Strip charts](#) show a plot of data values that InterMapper has recorded. These data may also be written to log files.

The [Debug Window](#) is normally hidden. It provides detailed information that may be useful when debugging either problems with the InterMapper program itself, or InterMapper's interactions with devices being tested.

Device Information Window

The device information window shows information about a device, and also allows you to configure the way *InterMapper* interacts with the device. Select the device and choose **Get Info...** from the **Network** menu, or double-click the device. Figure 5-1 shows a typical Device Information window.

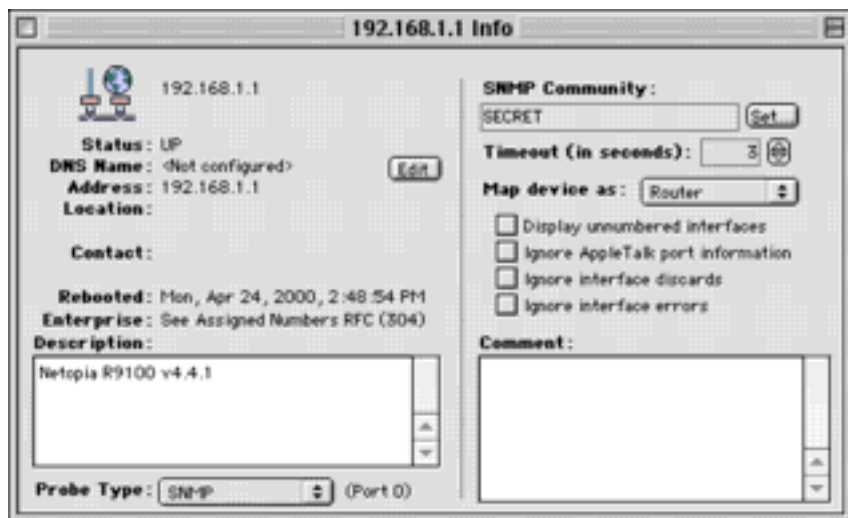


Figure 5-1: Device Info... window.

The window displays the following static information about the device:

- Status (UP, DOWN, ALARM, WARN)
- DNS Name and IP address or AppleTalk NBP name and address
- Location, Contact information, Uptime, and System description retrieved from SNMP queries.
- Probe type, as shown in the Probe Type pop-up menu.
- SNMP Read-only Community String to be used in SNMP queries.
- Device display parameters ("Map Device as:" pop-up menu, and associated checkboxes)
- A user-entered Comment: field, which can hold up to 255 characters of text regarding this device.

Changing a Device's Settings

The Device Information window also allows you to change the way *InterMapper* interacts with a device. The following parameters may be changed:

- Clicking the **Edit** button next to the **DNS Name:** field opens a window like Figure 5-2.

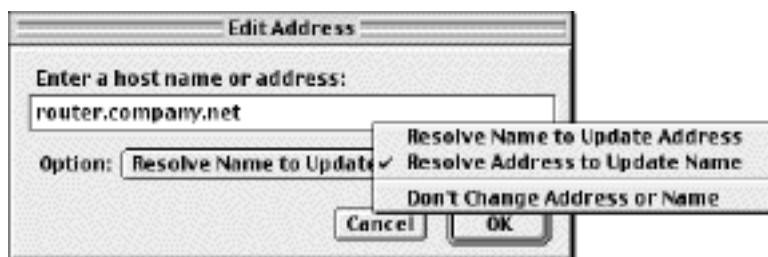


Figure 5-2: Use the Edit Address window to change the way *InterMapper* handles the DNS name or IP address for the device.

Enter a new DNS name or IP address here -- *InterMapper* will use this address when it probes the device. The **Option:** popup controls *InterMapper*'s automatic checking of DNS names and IP addresses. If you select **Resolve Name to Update Address**, *InterMapper* will query the DNS for the given name, and will change the address polled for the device. The **Resolve Address to Update**

- **Name** setting will keep the device's IP address fixed, but may update the name from the DNS server.
- Clicking the **Set...** button next to the **SNMP Community:** field will change the SNMP read-only community string used to query this device.
- **Map device as...** popup menu: By default, *InterMapper* attempts to determine the device type automatically. If *InterMapper* cannot correctly distinguish whether a device is a router or switch, it may be necessary to use this to select the proper choice. When set to a router, *InterMapper* determines network information for each port separately. For a switch, *InterMapper* propagates all network number information from one port to all ports. For the other settings (Hub and End System), *InterMapper* does not propagate network information about the ports.
- Check the **Ignore AppleTalk Port Information** box if *InterMapper* should not include AppleTalk network numbers for this device.
- Check the **Display Unnumbered Interfaces** box to see all the unnumbered interfaces on a switch. By default, *InterMapper* does not display unnumbered interfaces.
- Check the **Ignore Interface Discards** box to prevent discards from counting as errors. Certain equipment does not correctly report discards via SNMP, and causes the device to be shown in alarm.
- Check the **Ignore Interface Errors** box to prevent interface errors from marking the device as in alarm.
- The **Comment** field may contain a comment of up to 255 characters for this device. This information will be saved as part of the map, and might be used to save the model and serial number for a device, telephone numbers, circuit numbers, etc.

Selecting Probe Types

Use the **Probe Type** pop-up menu to select the kind of probe used to test the device. Each probe performs a different test on a device to verify that it working. Figure 5-3 shows the **Probe type** pop-up menu, and the choices available.

[Appendix A](#) describes *InterMapper*'s built-in probes which appear in this pop-up menu shown at the right. [Appendix B](#) describes how to create custom probes that will be added to this pop-up menu.

There are several sections of probes in the pop-up menu. They are categorized below:

- **Reload Probes...** This forces *InterMapper* to reload the probe files so that changes (after you edited a file) will become active. Select this and the files will be reloaded. You'll see a report of changes.
- **Intelligent probes.** When one of these is selected, *InterMapper* attempts to select the proper probe type for a device automatically.
 - Non-polling:** The device will not be probed, and will appear as a gray icon on the map.
 - Automatic:** *InterMapper* selects the probe automatically. The pop-up then switches to show the probe type actually used.
 - Ping/Echo:** *InterMapper* probes with ICMP or AppleTalk Echo packets, as is appropriate.
 - SNMP:** *InterMapper* queries the device with SNMP.
- **Custom SNMP Probes** are described in [Appendix B -- Custom SNMP Probes](#). The following probes are built-in: other custom SNMP probes that you create will also be listed here. You should also check the Contributions section of the *InterMapper* web site (<http://www.intermapper.com/contrib/>) for additional probes.
 - Basic OID:** Enter the name and OID of a MIB variable to monitor.
 - Cisco:** This probe monitors the % utilization and the internal temperature of a Cisco router.
 - TCP Check:** This probe puts the device into alarm if the number of TCP connections exceeds a certain threshold (default is 0). This could be useful for watching for intrusions.
- **UDP Probes.** These probes use UDP or proprietary protocols.

Reload Probes...
Non-Polling
● Automatic Ping/Echo SNMP
SNMP - Basic OID SNMP - Cisco SNMP - TCP Check
BlitzWatch DHCP/BOOTP Domain Name (DNS) KeyServer® Multicast Listener Network Time Radius RTMP
4D Server AppleShareIP Basic TCP Custom TCP CVS Server DND Protocol FileMaker PRO FirstClass Server FTP (Login) FTP (No Login) Gopher HTTP HTTP (Post) HTTP (Proxy) HTTP (Redirect) HTTPS HTTPS (Post) IMAP4 IRC LDAP LPR NNTP POP3 RTSP SMTP TELNET
Demo Probe

- BlitzWatch:** Tests a BlitzMail server.
- DHCP/BootP:** Tests a DHCP or BootP server.
- Domain Name Server:** Tests a DNS server.
- KeyServer :** Tests a KeyServer from Sassafras Software.
- Multicast Listener:** Tests a multicast stream.
- Network Time:** Tests an NTP server.
- Radius:** Tests a RADIUS authentication server.
- RTMP:** Sends AppleTalk RTMP Route Data Requests.
- **TCP probes.** *InterMapper* connects to a specified port on the remote device. A failure to establish a connection will indicate the device is down. If the connection completes, *InterMapper* interprets the device's response to determine the status (ok, down, warning or alarm).
 - 4D Server:** Makes a connection to a 4D database server and verifies that the specified database is available.
 - AppleShare IP:** Makes a connection to an AppleShare IP server.
 - Basic TCP:** Makes a TCP connection to a specified port on the device.
 - Custom TCP:** Connects to the device, sends a string, and matches the response to determine the device status.
 - CVS Server:** Connects to the server and authenticates using the supplied name and password.
 - DND Protocol:** Connects to a DND server (part of the BlitzMail system).
 - FileMaker Pro:** Makes a connection to a FileMaker Pro server.
 - FirstClass Server:** Makes a connection to a FirstClass e-mail server and verifies the banner.
 - FTP (Login):** Connects to FTP port, logs in with the specified name and password, and sends a NOOP command.
 - FTP (No Login):** Connects to FTP port and issues a NOOP command.
 - Gopher:** Makes a connection to the Gopher server on the device.
 - HTTP:** Retrieves a specified web page from the server.
 - HTTP (Post):** Posts data to an HTTP server.
 - HTTP (Proxy):** Tests an HTTP Proxy server
 - HTTP (Redirect):** Tests HTTP redirection.
 - HTTPS:** Retrieve a specified page from a secure web server over an SSL connection.
 - HTTPS (Post):** Posts data to an HTTPS server.
 - IMAP4:** Makes a TCP connection to an IMAP server on the device.
 - IRC:** Tests an Internet Relay Chat (IRC) server.
 - LDAP:** Tests an LDAP server.
 - LPR:** Tests an LPR server.
 - NTP:** Tests a Network Time Protocol (NTP) server.
 - NNTP:** Connects to a NNTP server and issues a GROUP command for the specified news group.
 - POP3:** Connects to a POP3 server, and verifies that the server sends "+OK".
 - RTSP:** Tests an Real Time Streaming Protocol (RTSP) server.
 - SMTP:** Connects to a SMTP server, and verifies a recipient exists with a VRFY command.
 - TELNET:** Connects to a Telnet server and authenticates with the supplied username and password.

Figure 5-3: Probe Popup

Choosing one of the Custom SNMP or TCP Probes opens a window that allows you to enter parameters for the probe, such as the remote port, the queries, the expected responses, etc. Figure 5-4 is an example of a TCP Probe configuration window. Double-clicking in the right-hand text box (for example, the one with "/") makes that text editable.



Figure 5-4: The Probe configuration window. This example shows the HTTP probe which has parameters for the remote port, the URL Path and the string that is expected in the response.

For more information about creating custom TCP probes, see [Appendix B -- Customizing InterMapper's Probes](#).

Network Information Window

InterMapper can display information about a network. Figure 5-5 shows the window which contains the list of networks that are defined on the particular link.

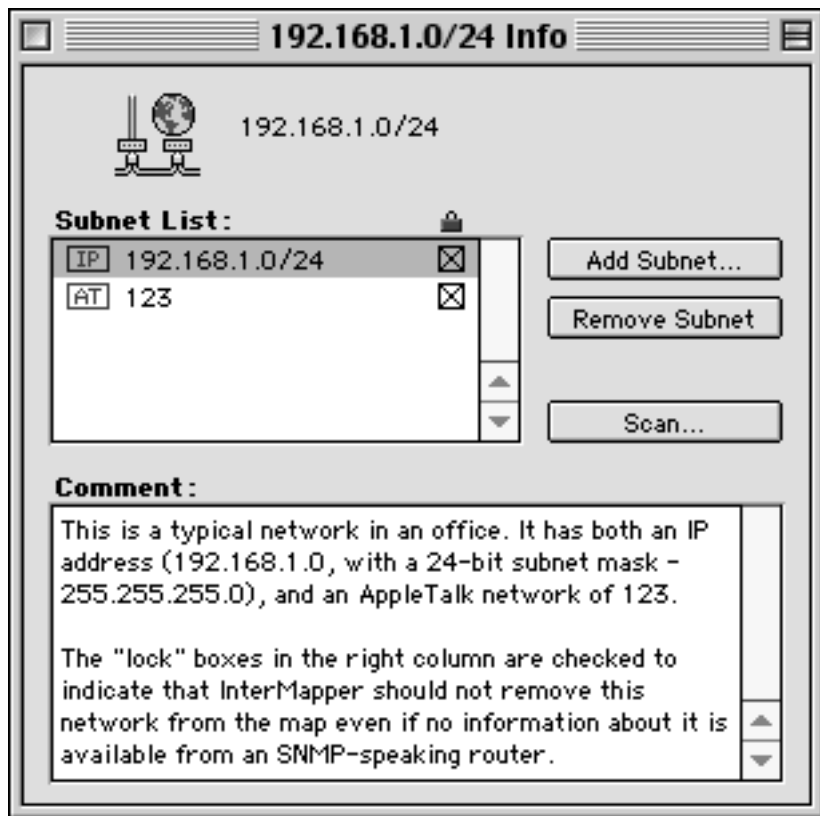


Figure 5-5: Network Information Window.

Each entry in this list has the following attributes:

IP or AT: Indicates whether the network is an IP or AppleTalk network.

Network ID: Gives the IP or AppleTalk network number for the network

Locked checkbox: This box indicates that the specific network ID will not be deleted. Automatically discovered networks (e.g., one connected to a SNMP-speaking router) will have this box checked by default. Networks that have been manually added (see below) must have this box checked to prevent them from being removed during *InterMapper*'s automatic "housecleaning".

Editing a Network

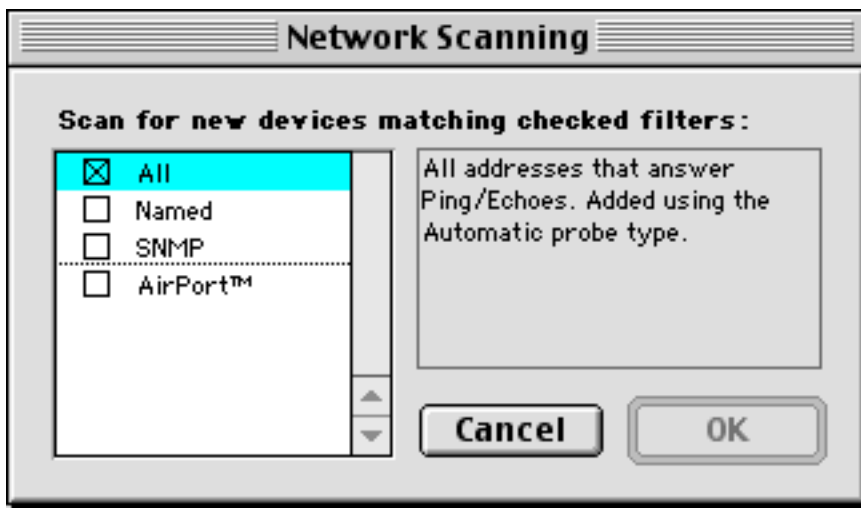
Add subnet...: Use this button to add a subnet to the list shown in this window in the event that *InterMapper* does not find that subnet automatically.

Remove subnet...: Use this button to remove a subnet from the list

Comment: Enter up to 255 characters as a comment about the network. This information is saved in the map, and could be used for a telco circuit number, location and contact information, etc.

Scanning a Network

The **Scan...** button causes *InterMapper* to scan the selected network(s) for devices and add new ones to the map. This button opens the window of Figure 5-6 to allow you to specify the kinds of devices to be added.



This window allows you to specify which kinds of devices will be discovered during InterMapper's initial scan of the network. Checking any of the boxes will make InterMapper look for the particular device type. Clicking on a device type (but not in the checkbox) will display a description of the device type.

Figure 5-6: Network Scanning Window.

Event Log Windows

InterMapper writes information about interesting events into *event logs*. These streams of information are written to log files on-disk, and also displayed in windows on the screen. These *event log windows* allow the user to review the past events without resorting to an external text editor to review the log files.

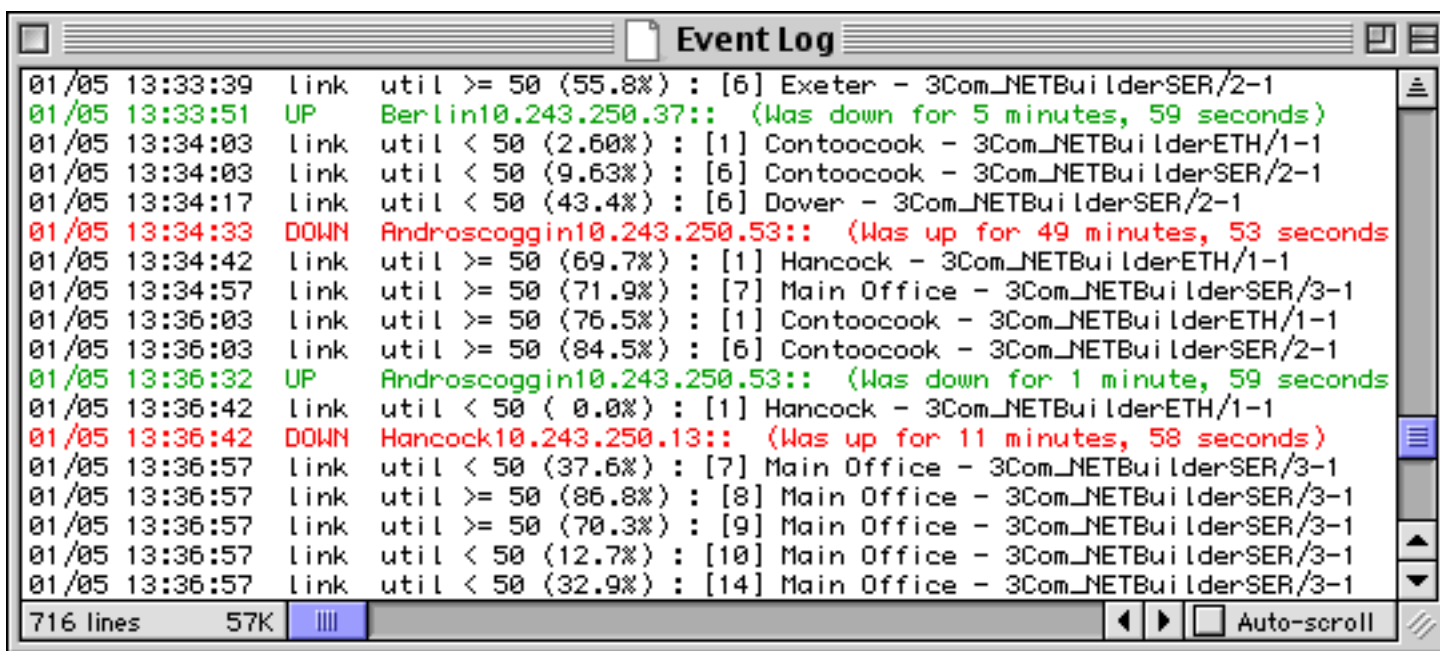


Figure 5-7: The main Event Log window. It can show information about device ups and downs, high traffic on links, web, telnet, and InterMapper Remote server connections, as well as error messages.

Event Log Controls

This window lists the events that have occurred recently. They are listed chronologically, with the newest at the bottom. Clicking the **pyramid icon** at the upper right (above the scroll bar) will change the display order, so the newest are shown at the top.

By default, InterMapper always scrolls the data of the window so the newest entry is visible. Un-checking the **Auto-scroll** box freezes the display, so that new entries won't move the view. You can still scroll around the entries, but they will not move, even if new entries appear in the log file. They will, of course, be written to the log file and to the data in the window.

Redirecting Log Streams

By default, all event log entries are written to the main Event Log window/file. However, these streams of event information may be configured to go to different log windows/files.

For example, the Web, Telnet, and InterMapper Remote server access messages may not be as interesting or important, and might clutter the main log file. Therefore, a network manager can configure these servers to send their messages to separate log files. Figure 5-8 below shows a log file's window that receives Web, Telnet, and InterMapper Remote session information.

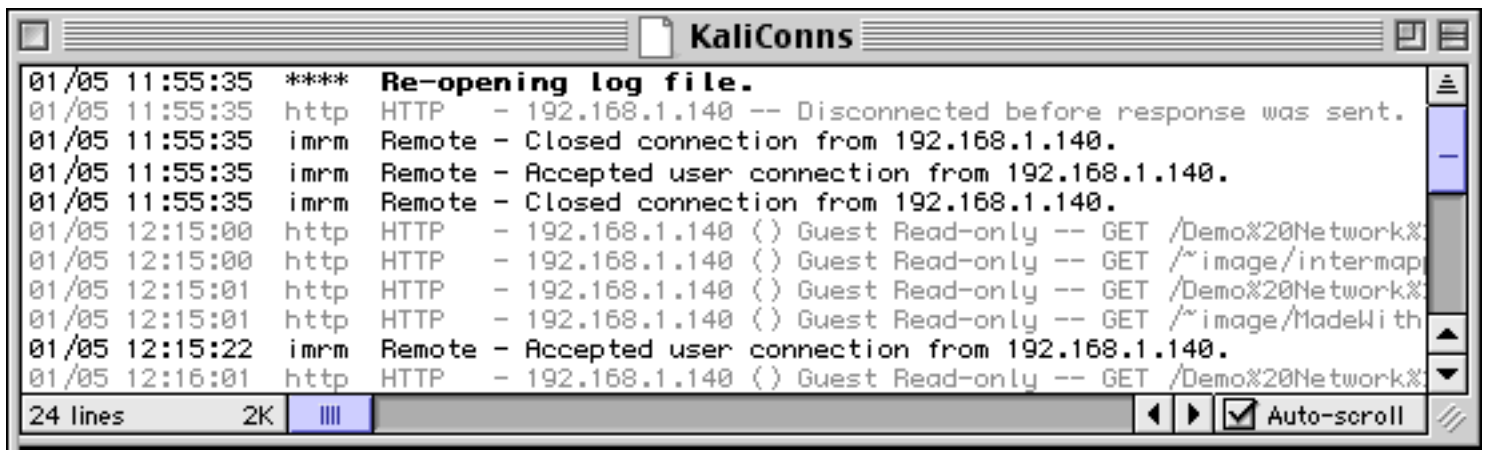


Figure 5-8: An event log window showing Web (*http*) and InterMapper Remote (*imrm*) sessions.

The [Log File Preferences](#) of Chapter 4 discusses the details of creating new log files and redirecting log streams to those separate files. s

Error Messages in the Log Windows

All log windows window display lines note events that have occurred. InterMapper writes these events to various kinds of log windows: Device Messages have the following format:

```
<timestamp> <tag> <fullname>:: <message>
```

The <tag>s are:

```
"UP " : <message> = "(Was down for <duration:3>)" or ""
"DOWN": <message> = "(Was up for <duration:3>)" or ""
"okay": <message> = <threshold-condition>
"warn": <message> = <threshold-condition>
"alm": <message> = <threshold-condition>
"ACK " : <message> = <acknowledge-message>
"UNAC": <message> = ""
"TRAP": <message> = <trap-message>
```

where the <duration:3> can be one of:

- "[0-9]+ seconds?"
- "[0-9]+ minutes?, [0-9]+ seconds?"
- "[0-9]+ hours?, [0-9]+ minutes?, [0-9]+ seconds?"
- "[0-9]+ days?, [0-9]+ hours?, [0-9]+ minutes?"

All other log lines have the format:

```
<timestamp> <tag> <message>
```

List of log messages

This page lists all the error log messages that InterMapper displays, with a description of each. Items shown in *italics* are variable names that are substituted with the proper value when the log message is created.

As of this date, the set of descriptions is incomplete. Please send a note to support@dartware.com if you have questions about any of these messages or descriptions.

General Messages

These messages describe InterMapper's actions as it starts up, enables and disables servers, and opens and closes the map files. These messages always go to to the Event Log window.

**** Starting *appName*

InterMapper is starting up. This entry contains the program's version number

**** Quitting *appName*

The InterMapper application is quitting.

**** Opening map *docName*

The named map is being opened.

**** Closing map *docName*

The named map is being closed.

http Starting web server on port *portnumber*

InterMapper is starting its web server on port *portnumber*

http Stopping web server on port *portnumber*

InterMapper is stopping the web server

kali Starting KALI server on port *portnumber*

InterMapper is starting its InterMapper Remote ("Kali") server on port *portnumber*

kali Stopping KALI server on port *portnumber*

InterMapper is stopping the InterMapper Remote server

tlnt Starting telnet server on port *portnumber*

InterMapper is starting its Telnet server on port *portnumber*

tlnt Stopping telnet server on port *portnumber*

InterMapper is stopping the Telnet server

Start-up error: Could not open *porttype* port *portnumber*.

InterMapper could not open the specified port during startup

kOTLookErr sending udp packet. (OTLook() = *errNumber*)

InterMapper received an error attempting to send a UDP packet

Error sending udp packet. (Err = *errNumber*)

InterMapper received an error attempting to send a UDP packet

DNS-Related Messages

**** Address Change: "www" changed from x.x.x.x to y.y.y.y. (DNS z.z.z.z)

The Device named www changed its IP address from x.x.x.x to y.y.y.y according to the DNS server at z.z.z.z

**** Address Change: "www" changed from x.x.x.x to y.y.y.y.

The device named www changed its IP address from x.x.x.x to y.y.y.y.

**** No IP address for "www". (DNS z.z.z.z)

InterMapper was not able to determine an IP address for the device named www from the DNS server at z.z.z.z

**** Name Change: "w.w.w.w" changed from "xxx" to "yyy". (DNS z.z.z.z)

The IP address w.w.w.w changed its DNS name from xxx to yyy according to the DNS server at z.z.z.z

**** No domain name for x.x.x.x. (DNS z.z.z.z)

InterMapper was not able to determine a DNS name for x.x.x.x from the DNS server at z.z.z.z

**** "No response from DNS x.x.x.x when resolving 'yyy' to an address.

The DNS server at x.x.x.x did not respond when attempting to resolve the DNS name yyy to an address.

**** "No response from DNS x.x.x.x when resolving 'y.y.y.y' to a name.

The DNS server at x.x.x.x did not respond when resolving the address y.y.y.y to a name.

debug "DNS packet with bad format from %#s", addrStr

needs_text_entry_here

debug "Error %ld while processing DNS reply from %#s", error, addrStr

needs_text_entry_here

Probe File Error Messages

These messages describe problems with the Custom Probe files.

debug "%#s: Can't match \"%#s\"", cFileName, lineStr

needs_text_entry_here

debug "%#s: Can't match \"%#s\"", cFileName, lineStr

needs_text_entry_here

debug "%#s: Invalid Probe ID.", cFileName

needs_text_entry_here

debug "%#s: Invalid Probe Name.", cFileName
needs_text_entry_here

debug "%#s: Invalid Probe Human Name.", cFileName
needs_text_entry_here

debug "%#s: Probe definition does not contain a valid <description> section.", cFileName
needs_text_entry_here

debug "%#s: Probe definition does not contain a valid <snmp-device-variables> section.", cFileName
needs_text_entry_here

debug "%#s: Probe definition does not contain a valid <snmp-device-display> section.", cFileName
needs_text_entry_here

debug "%#s: Probe definition does not contain a valid end tag for <%#s>.", cFileName, endTagStr
needs_text_entry_here

debug "%#s: Can't match \"%#s\"", fileName, lineStr
needs_text_entry_here

Telnet Server Messages

**** x.x.x.x denied access to tcp server.
An attempt to connect to the Telnet server from address x.x.x.x was refused.

tlnt TELNET - x.x.x.x denied access.
An attempt to connect to the Telnet server from address x.x.x.x was refused.

tlnt TELNET - x.x.x.x denied access because there are too many connections.
An attempt to connect to the Telnet server from address x.x.x.x was refused because there were too many connections already established.

tlnt TELNET - Accepted connection from x.x.x.x
A user at x.x.x.x successfully connected to the Telnet server.

tlnt TELNET - Accepted user connection from x.x.x.x
Telnet server accepted a user connection from x.x.x.x

tlnt TELNET - x.x.x.x authenticated as "username".
The Telnet server accepted a connection from an authenticated user, "username"

tlnt TELNET - Closed connection from x.x.x.x
The user at address x.x.x.x disconnected from the Telnet server.

Trap-Related Messages

trap "%#s (not on map) :: %#s", addrStr, msgStr
needs_text_entry_here

trap "An error occurred while processing a SNMP trap from %#s. (err = %ld)", addrStr, (long) err
needs_text_entry_here

Notification Messages

ntfy "Silenced email notification to \"%#s\".", itsUserName
needs_text_entry_here

ERR! "Failed to send email notification to \"%#s\" for \"%#s: %#s\" event. Check email configuration. (err = %d)", itsUserName, eventMsg, deviceName, err
needs_text_entry_here

ntfy "Sent email notification to \"%#s\" for \"%#s: %#s\" event. (%#s of %#s)", itsUserName, eventMsg, deviceName, sendCount, maxSendCount
needs_text_entry_here

ntfy "Silenced pager message notification to \"%#s\".", itsUserName
needs_text_entry_here

ERR! "Failed to send pager notification to \"%#s" for "%#s: %#s". (err = %d)", itsUserName, eventMesg, deviceName, err
needs_text_entry_here

ntfy "Sent pager message notification to \"%#s" for "%#s: %#s\".", itsUserName, eventMesg, deviceName
needs_text_entry_here

ntfy "Silenced apple event notification to \"%#s\".", itsUserName
needs_text_entry_here

ERR! "Failed to send apple event notification to \"%#s" for "%#s: %#s". (err = %d)", itsUserName, eventMesg, deviceName, err
needs_text_entry_here

ntfy "Sent apple event notification to \"%#s" for "%#s: %#s\".", itsUserName, eventMesg, deviceName
needs_text_entry_here

**** "Silenced sound notification to \"%#s\".", itsUserName
needs_text_entry_here

ERR! "Failed to send sound notification to \"%#s\". (err = %d)", itsUserName, err
needs_text_entry_here

ntfy "Silenced SNMP trap notification to \"%#s\".", itsUserName
needs_text_entry_here

ERR! "Failed to send SNMP trap notification to \"%#s" for "%#s: %#s". (err = %d)", itsUserName, eventMesg, deviceName, err
needs_text_entry_here

ntfy "Sent SNMP trap notification to \"%#s" for "%#s: %#s\".", itsUserName, eventMesg, deviceName
needs_text_entry_here

ERR! "Failed to send email notification to %#s. Check user configuration.", itsUserName
needs_text_entry_here

ERR! "Failed to send pager notification to %#s. (err = %d)", itsUserName, err
needs_text_entry_here

**** "Silenced all notifications until %#s.", timeStr
needs_text_entry_here

kERR! "Can't locate target of apple event notification. err =%d", err
needs_text_entry_here

kERR! "Can't launch or communicate with target of apple event notification. err = %d", err
needs_text_entry_here

ERR! "SMTP Failure: Can't connect to \"%#s\". Error = %d", itsMailServer, err
needs_text_entry_here

ERR! "SMTP Failure: Server connection to \"%#s" idle for more than 4 minutes. Disconnecting...", itsMailServer
needs_text_entry_here

ERR! "SMTP Failure: Server \"%#s" won't accept mail from %#s. (Reply = %d)", itsMailServer, reversePath, replyCode
needs_text_entry_here

ERR! "SMTP Failure: Server \"%#s" rejected recipient %#s. (Reply = %d)", itsMailServer, emailAddr, replyCode
needs_text_entry_here

ERR! "SMTP Failure: Server \"%#s" failed when sending mail to %#s. Mail not sent. (%s Reply = %d)", itsMailServer, emailAddr, cmdName, replyCode
needs_text_entry_here

Web Server Messages

http HTTP - *address (user) authLevel -- command argument*

InterMapper received a *command* request for *argument* from *address*

http HTTP - ERROR: JPEG compression failed. Image too wide (width > 4096 pixels). Try QT4 + MacOS 8.6.

InterMapper was unable to create a JPEG image. Try using QuickTime 4 and MacOS 8.6 or newer.

http HTTP - ERROR: JPEG compression failed. Compressed length = xxx. (Error = yyy)

InterMapper got an error code of yyy when attempting to compress the JPEG image whose length is yyy bytes.

http HTTP - ERROR: JPEG compression failed. Can't obtain/lock PixMap

InterMapper was unable to compress a JPEG image because it was already compressing an image. If this problem persists, quit InterMapper and relaunch it.

http HTTP - ERROR: JPEG compression failed. Can't create graphics offscreen. (Error = yyy)/dt>

InterMapper received the yyy error code when attempting to compress a JPEG image.

http HTTP - ERROR: PNG compression failed because there is not enough memory. (yyy K available)

InterMapper failed to compress a PNG

http "HTTP - ERROR: PNG compression failed. (Error = %d)", err

needs_text_entry_here

http "HTTP - %#s -- Unknown HTTP Version: %s", addrStr, vers

needs_text_entry_here

http "HTTP - %#s -- Missing HTTP Version.", addrStr

needs_text_entry_here

http "HTTP - %#s -- Unknown HTTP Command: %s", addrStr, cmd

needs_text_entry_here

http "HTTP - %#s -- Disconnected before response was sent.", addrStr

needs_text_entry_here

http "HTTP - ERROR: JPEG image cannot be generated because QuickTime is not installed."

needs_text_entry_here

http "HTTP - ERROR: JPEG image cannot be generated because the Image Compression Manager is not installed."

needs_text_entry_here

http "HTTP - ERROR: JPEG image cannot be generated because QuickTime PowerPlug is not installed."

needs_text_entry_here

http "HTTP - ERROR: Unable to create %ld x %ld JPEG image. (Error = %d)", pictWidth, pictHeight, err

needs_text_entry_here

http "HTTP - ERROR: Unable to create %ld x %ld PNG image. (Error = %d)", pictWidth, pictHeight, err

needs_text_entry_here

link "%#s (%#s%%) : [%ld] %#s - %#s"

needs_text_entry_here

ants "%#s Incoming : [%ld] %#s - %#s"

needs_text_entry_here

ants "%#s Outgoing : [%ld] %#s - %#s"

needs_text_entry_here

debug "%#s UTIL[%lu]=%#s? %#s: upTimeNow=%lu, upTimePrev=%lu; inOctetNow=%lu, inOctetPrev=%lu; outOctetNow=%lu, outOctetPrev=%lu; bps=%lu

needs_text_entry_here

debug Saved backup copy of *mapname* in "InterMapper Settings:Old Maps" folder.

InterMapper saved a copy of the original file (*mapname*) in the Old Maps folder before saving a version of the file in a newer format. This allows you to retrieve the earlier file and use it with an older

copy of InterMapper.

dbug An error occurred while attempting to save backup copy of *mapname*

InterMapper was not able to create a backup copy of the named map

dbug Can't locate backup folder to save backup copy of *mapname*

InterMapper couldn't locate or create the InterMapper Settings:Old Maps folder.

dbug Device '*devicename*' was using non-existent probe '*probename*', now set to non-polling.

The named device was set to be probed with a non-existent probe type. It has been set to "non-polling", and will no longer be probed.

InterMapper Remote ("kali") Server Messages

kali "KALI - Accepted user connection from %#s.", addrStr
needs_text_entry_here

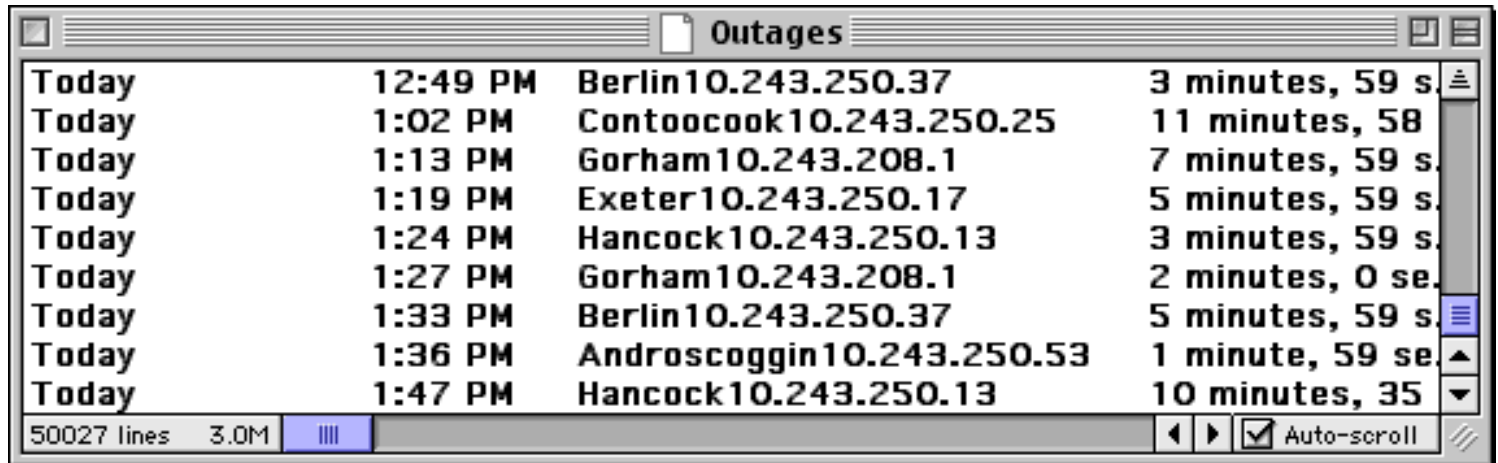
kali "KALI - Closed connection from %#s.", addrStr
needs_text_entry_here

kali "KALI - %#s denied access.", addrStr
needs_text_entry_here

kali "KALI - %#s denied; too many connections.", addrStr
needs_text_entry_here

Outages Window

InterMapper summarizes outages that have occurred in the **Outages Window**. An *outage* is defined as a device that has gone from the UP state to the DOWN state, and then returned to the UP state. InterMapper tracks the start and end time of the outage, and computes the duration



Time	Device Name	Duration
Today 12:49 PM	Berlin10.243.250.37	3 minutes, 59 s
Today 1:02 PM	Contoocook10.243.250.25	11 minutes, 58
Today 1:13 PM	Gorham10.243.208.1	7 minutes, 59 s
Today 1:19 PM	Exeter10.243.250.17	5 minutes, 59 s
Today 1:24 PM	Hancock10.243.250.13	3 minutes, 59 s
Today 1:27 PM	Gorham10.243.208.1	2 minutes, 0 se.
Today 1:33 PM	Berlin10.243.250.37	5 minutes, 59 s
Today 1:36 PM	Androscoggin10.243.250.53	1 minute, 59 se.
Today 1:47 PM	Hancock10.243.250.13	10 minutes, 35

50027 lines 3.0M Auto-scroll

Figure 5-8: The Outages window shows the start and end time and the duration of outages.

Note that the "Androscoggin" outage listed in this window matches the DOWN and UP messages shown in Figure 5-8 on the [Event Log](#) page.

The controls in the in the Outages window are identical to those of the [Event Log](#) windows, and are described on that page.

Strip Chart Windows

InterMapper displays historical information in a *strip chart*. Strip charts can hold an unlimited number of data sets for an unlimited time period. These data can also be written to a tab-delimited text file.

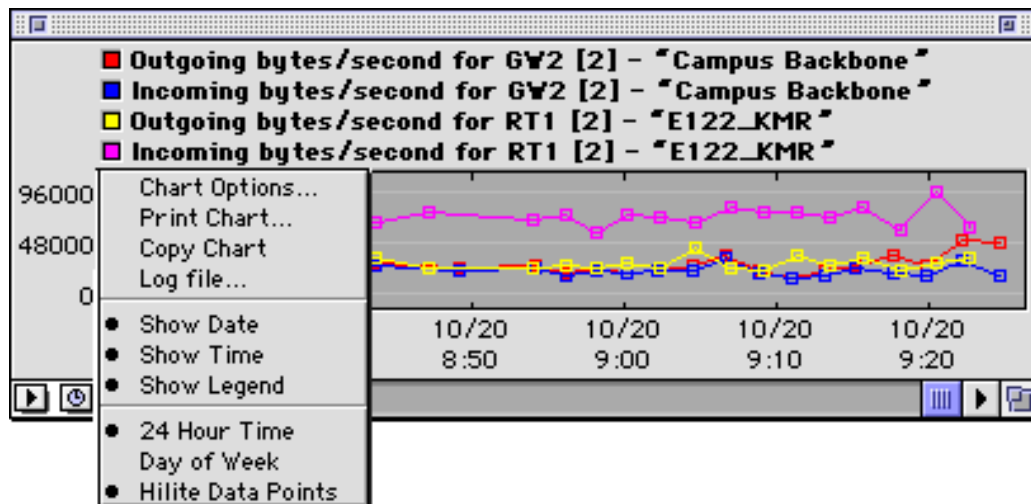


Figure 5-10: A strip chart showing four traces.

The figure above shows a strip chart with four variables. To **add** a variable to an existing chart, drag an underlined value (blue or grey) from a popup window into the strip chart. To **remove** a variable from an existing chart, click and drag its color rectangle to the Finder's trash.

Strip charts automatically save their data to disk in the *InterMapper Settings:Chart Data* folder. When *InterMapper* launches, it reads the data from these files to restore the strip chart contents.

Strip Chart Popup Menus

The pop-up menus at the lower left corner control the chart's labels, options, and time intervals. The items in the popup menu include:

Chart Options... configures the strip chart's title, its vertical scale, and the name of a log file which will hold the logged data. Setting a chart's title also changes how the chart is displayed in the *InterMapper* web interface. These are described in detail below.

Print Chart... prints the chart, using the standard Print... dialog.

Copy Chart... copies the chart as a picture (PICT format) to the Clipboard, for pasting in other documents.

Log file... specifies a log file to receive copies of the data. See the section below on chart log files.

Show Date When checked, the strip chart will include dates at the bottom of the chart.

Show Time When checked, the strip chart will include the time at the bottom of the chart.

Show Legend When checked, the strip chart will include the name of the data variable (or the *chart legend* for [Custom SNMP probe variables](#)) at the top of the chart.

Day of Week When checked, the strip chart will include the day of the week below the chart.

24 Hour Time When checked, the strip chart will display data in a 24-hour clock.

Hilite Data Points When checked, each data point will have a small square drawn on it; when unchecked, the strip chart will simply show line segments.

Clock icon Select the horizontal time scale for the strip chart by clicking on the clock icon at the lower left. You may choose a time interval between one minute and 24 hours.

Strip Chart Options

Strip charts have a number of options that are controlled by the Chart Options window, shown in the figure at the right. Open this window by selecting **Chart Options...** from the chart's popup menu.

The **Title:** sets the text in the title bar of the strip chart's floating window.

The **Vertical Axis** items control the upper and lower Y-axis labels on the strip chart, the number of horizontal dividers in the chart, and whether the axis labels should auto-adjust.

Note: Auto-adjust sets the vertical scale so that the largest data value that is visible in the window will be near full-scale. Scrolling back or forward may cause data values to be trimmed. If this occurs, wait until the next data sample is added to the chart, and the axes will be re-drawn.

The **Legend & Color and Text** list shows the current color and text legend associated with each variable. You may re-order the items in this list by dragging them to the proper position. You can edit the color or text by double-clicking an item. A window like Figure 5-12 below will open.

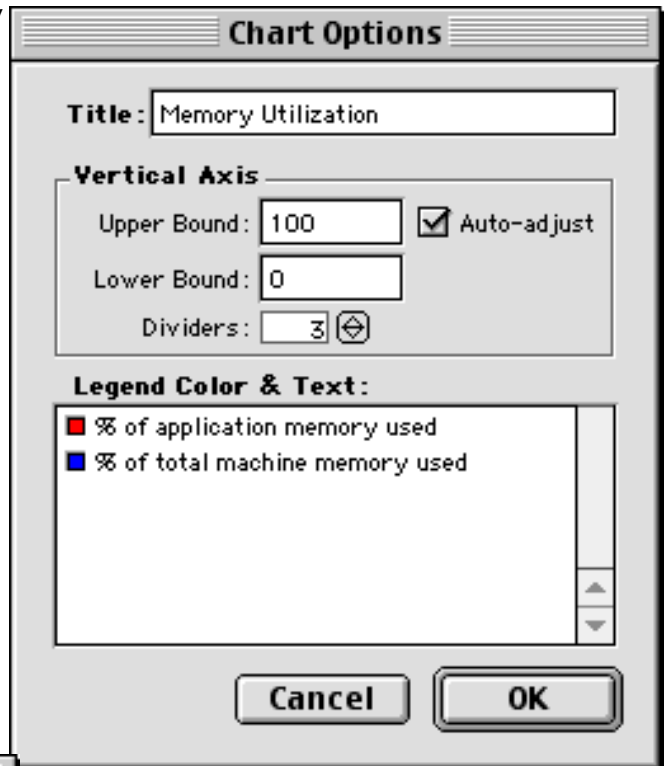


Figure 5-11: Chart options for a strip chart.



Figure 5-12: Editing Chart Colors and Text

Click the colored rectangle to get a standard MacOS color picker. Select a line style or width from the **Style:** popup menu. Edit the text of the legend in the **Title:** field.

Strip Chart Log Files

InterMapper will also write strip chart data to a tab-delimited text file. Each variable will be written to its own column of data. You can specify a log file to receive the data by selecting **Log file...** from the strip chart popup menu. The specification window resembles Figure 5-13 at the right.

For more details about the log file editing dialog, see the [Logging Prefs](#) page.

To cease logging data to a log file, go to the [Logging Preferences](#) and delete the log file. Strip chart data will no longer be written to that file.

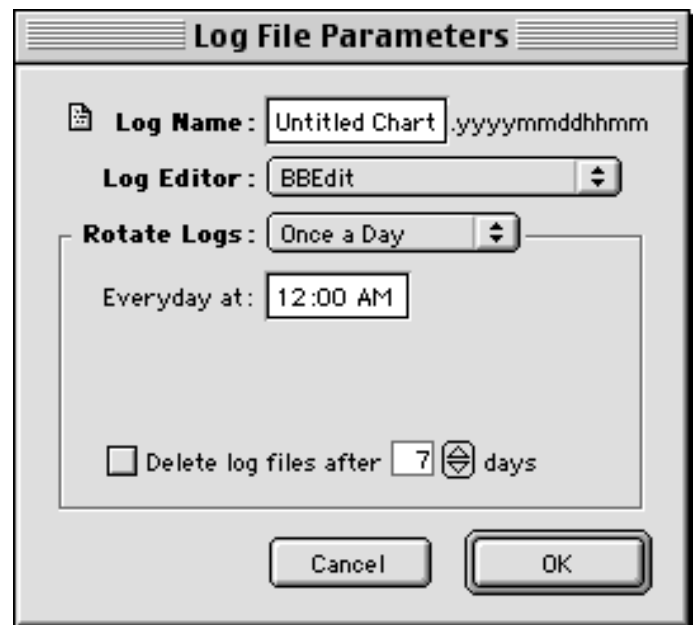


Figure 5-13: Selecting a strip chart log file

Debug Window

InterMapper has a facility for displaying debugging information, called the "Debug window". Its contents generally won't need to be observed by the majority of InterMapper users, and consequently, it requires a special command-key sequence to open it.

The Debug window contains messages show details of InterMapper's operations that can be valuable for debugging problems with the program. If you have trouble with InterMapper, the support staff may ask you to open the debug window, turn on certain options, and read or e-mail the contents of the window to us.

How to open the Debug Window

To open the Debug window, hold down Command, Option, and Shift, then press the "Z" key. You'll see the Debug window open, and a new **Debug** menu appear in the menu bar at the top of the screen.

The entries that are added to the Debug log window will also be written to the Debug log file. If the size of the Debug log file becomes a concern, you can use the Logging Prefs to have InterMapper rotate the log files and delete them daily.



In general, Dartware does not document the information shown in the Debug window, because its messages will change from version to version.



Note: Certain of the items in the **Debug** menu are designed to test InterMapper's crash recovery facilities. Certain others may exercise portions of the program that may crash.

Built-in Web Server

InterMapper can act as a web server, publishing most of the information that is available from the program. Before *InterMapper* will accept web connections, you must configure the web preferences as described in the **Access Control for Web Pages** section, below.

Each page that *InterMapper* serves contains the same controls at the top. Figure 6-1 shows a typical web page.

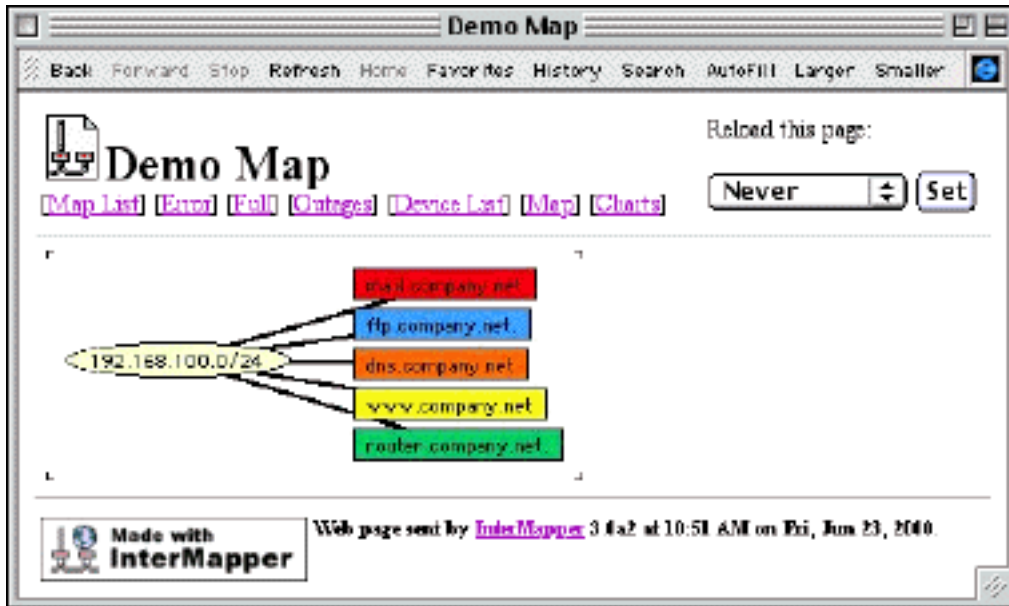


Figure 6-1: A typical *InterMapper* web page. This example shows information about the "Demo Map".

The *InterMapper* web page typically has three parts:

- The *header* which shows the map name or other title, a navigation bar for going to other pages, and the refresh popup menu. This is usually the same for every page.
- The *content* of the page, which gives the information of the page
- The *footer* of the page which shows the time the page was created.

Web Page Header

The Web Page Header displays a page title, and provides navigational links. Figure 6-1 shows a web page for a particular map; Figure 6-2 below shows the navigational links for the global pages.



Figure 6-2: The header for the global web pages. Note that the navigational links list different choices from a map's links.

Clicking on any of the standard navigational links will lead to one of *InterMapper's* seven informational pages. These are described in a separate pages.

Reloading the Page

InterMapper pages can be set to reload the page at a specified interval. This will provide up-to-date information on the web page. To do this, choose a time period from the **Reload this page:** pop-up menu and click the Set button to cause the web browser to refresh the page at the interval indicated.

Customizing Web Pages

The appearance of InterMapper's web pages is controlled by template files saved on the hard drive. See [Appendix C -- Customizing Web Pages](#) for more information about editing these templates.



Controlling Access to Web Pages

You can set up access control lists to allow or deny access to certain web pages. See [Chapter 4 -- Web and Telnet Preferences](#) for more information about controlling access to the web services.

The Error and Full Pages

The **Error** page displays the devices that are down, or in alarm or warning states. It is the default page that appears when a browser first connects to *InterMapper*.

The **Full** page displays all the devices that are being monitored by InterMapper, not just those with problems. Both the **Error** and **Full** page have the following format, shown in Figure 6-3.

Nov 24 14:58:12 162 nodes, 1 down, 348 links, 4 down, 8283 pk/s, 2268 K by/s

Device	Stat	SysUpTime	Avail	Loss	Probe	Address
valley5	DOWN	0+00:01:38	63.5	63.2	SNMP	198.115.160.180

Link	Prt	Stat	TPkt	TBytes	TErr	TDis	RPkt	RBytes	RErr	RDis	Util	Segment
Burke (2)	A10	UP	13	1941	0	0	11	268	4*	0	7.7%	
Rope Ferry (7)	A10	UP	8	2564	0	0	7	437	2*	0	10.4%	
Silsby1 (10)	A10	UP	11	2636	0	0	10	416	1*	0	10.6%	
Fairchld1 (12)	A10	UP	16	2048	0	0	16	455	6*	0	8.7%	
Dartrow (14)	A10	UP	7	1681	0	0	7	320	1*	0	7.0%	
Blunt (15)	A10	UP	2	656	0	0	1	30	1*	0	2.4%	
Fleet (20)	A10	UP	17	1549	0	0	42	19874	40*	0	74.4%*	
The Hop (21)	A10	UP	5	811	0	0	5	111	1*	0	3.2%	
McNutt1 (55)	A10	UP	22	2310	0	0	20	401	12*	0	9.4%	
College (57)	A11	UP	12	2795	0	0	11	383	1*	0	11.0%	
Gym (62)	A10	UP	8	2278	1*	0	8	334	3*	0	9.1%	

Figure 6-3: The InterMapper Errors or Full web page. Note that the "reason" for the device being listed is shown in red.

The top line contains this information:

- date and time the page was generated
- the number of devices being monitored
- number of devices currently shown as down
- number of links being monitored
- total packets per second entering the network
- total bytes per second entering the network.

Following that information, there is a section describing **devices** that are down, or in alarm or warning states. This section contains:

- device's name (which is a link to more information)
- the device's status
- its uptime (sysUptime)
- availability
- packet loss
- probe type
- network address.

Next comes a section showing networks and links that are down, or in alarm or warning states. The page displays:

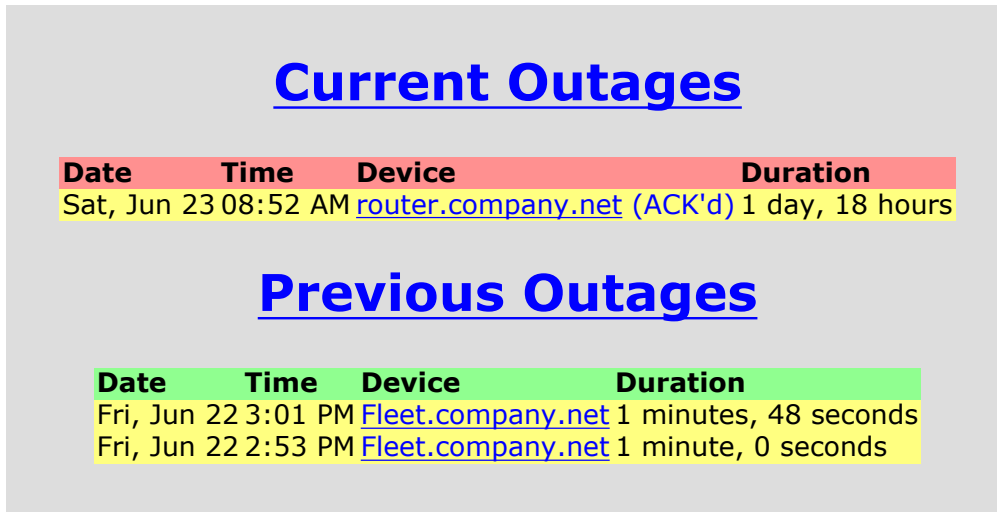
- the name of the device and port number
- its status
- transmit information (transmitted packets/second and bytes/second, transmit errors per minute, transmit discards per minute)
- receive information (packets, bytes, errors and discards)

- network utilization
- segment name (if any).

Clicking on any device or network links will open a page showing the same information as the pop-up window on the main InterMapper screen. Figure 6-7 and 6-8 on the [map web page](#) show typical Device and Network information pages.

The Outages Web Page

The **Outages** link provides a historical list of devices that have been down. Figure 6-4 shows a sample. Clicking on a link leads to detailed information similar to those on the [Web Map page](#).



The screenshot shows a web page with two sections: 'Current Outages' and 'Previous Outages'. Each section has a table with columns for Date, Time, Device, and Duration. The 'Current Outages' table has one entry for 'router.company.net (ACK'd)' with a duration of 1 day, 18 hours. The 'Previous Outages' table has two entries for 'Fleet.company.net' with durations of 1 minute, 48 seconds and 1 minute, 0 seconds.

<u>Current Outages</u>			
Date	Time	Device	Duration
Sat, Jun 23	08:52 AM	router.company.net (ACK'd)	1 day, 18 hours

<u>Previous Outages</u>			
Date	Time	Device	Duration
Fri, Jun 22	3:01 PM	Fleet.company.net	1 minutes, 48 seconds
Fri, Jun 22	2:53 PM	Fleet.company.net	1 minute, 0 seconds

Figure 6-4: The InterMapper Outages web page.

This page lists the last 10 outages for each device.

The Device List Web Page

The **Devices List** link displays a page that lists the device's status, name, condition, and date and time of the last change in status. Clicking on a link displays detailed information such as Figure 6-5.

Device List for "Demo Map"






Status	Name	Condition	Date Time
	mail.company.net	Down	06/23 10:48:06
	ftp.company.net	Down	06/23 10:49:32
	dns.company.net	[DNS] IP address in response doesn't match "129.170.16.79"	06/23 10:47:59
	www.company.net	[HTTP] "" not found in returned HTTP data.	06/23 10:48:02
	router.company.net	OK	06/23 10:47:48

Figure 6-5: Web Device List. This shows the status of all the devices InterMapper is monitoring, sorted by severity of their status.

The Map Web Page

The **Map** web page displays a snapshot of a particular map at the moment the page was produced.

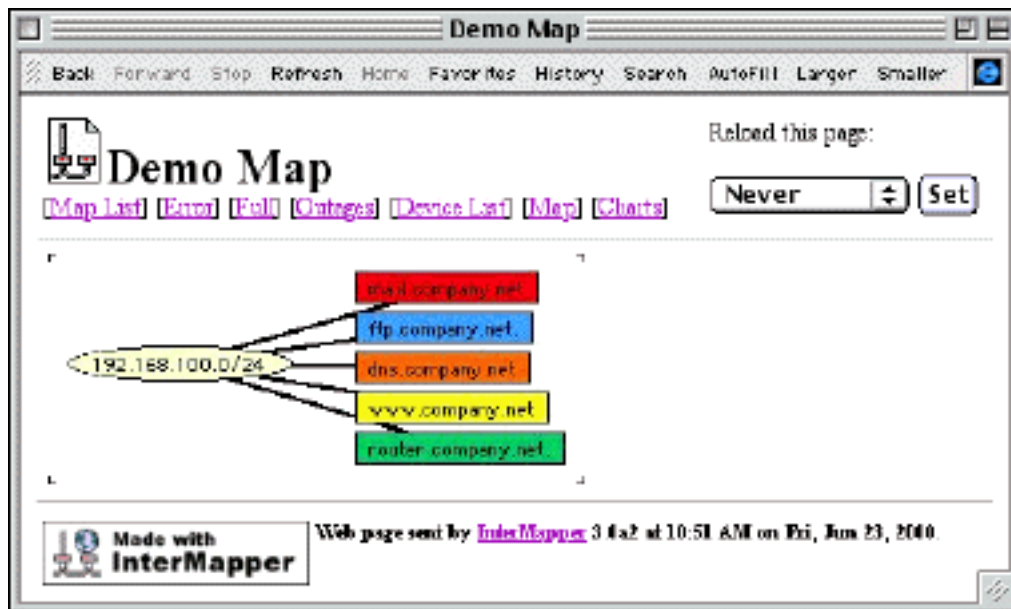


Figure 6-6: A typical Map page. Clicking on links, networks or devices produces a page with a detailed view of the item's information.

Each map page contains an image of the map shown on the server's screen. *InterMapper* will generate a PNG (Portable Network Graphics) if the web browser requests that format, or JPEG (Joint Photographers Experts Group) image otherwise. These images are static, but can be updated periodically using the **Reload this page:** pop-up.

Clicking on any of the devices, links, or networks shown in the Map window will produce a web page that shows detailed information for that item. This is the same information as in the pop-up window on the *InterMapper* screen. Here are typical displays:

Device Information

```
Device Information
Name:      router.company.net.
DNS Name:  router.company.net.
Address:   192.168.1.1
Status:    UP
Protocol:  Ping/Echo
Up Time:   n/a
Availability: 100 % (of 1 hour, 29 minutes, 12 seconds)
Packet Loss: 0.0 % (of 143 total attempts) [Reset]
Recent Loss: None
Last updated Jun 23, 12:16:42; interval: 30 seconds
```

Figure 6-7: Typical Device Information.

Network Information

Network Information

```
Name: 192.168.1.0/24
IP Net: 192.168.1.0/24 (255.255.255.0)
AT Net: 123
Sum In: 2 pkt/sec 548 byte/sec 0 error/min
Sum Out: 3 pkt/sec 316 byte/sec 0 error/min
```

Comment:

This is the network in the office. It has both an IP address (192.168.1.0, with a subnet mask of 255.255.255.0) and AppleTalk network number 123.

Figure 6-8: Typical Network information.

Link Information

Interface Information (ifIndex = 1)

```
Device Name: router.company.net.
Description: EN1
Type: 10 MBit ethernetCsmacd (MTU=1500)
Status: UP for 4 days, 13 hours
Address: 192.168.1.1 (255.255.255.0)
MAC Address: 00-00-C5-76-E2-EC
```

Interface Statistics

```
Utilization: 0.01 % (of 10 MBit bandwidth)
Percent Err: 0.0 % (59 pkts w/o error)
Transmit Statistics (0.01 % utilization)
  Pkt/Second: 0 (5.88 % multicast)
  Byte/Second: 73 (590 bps)
  Err/Minute: 0 (0 errors)
  Disc/Minute: 0 (0 discards)
  Percent Err: 0.0 % (17 pkts w/o error)
Receive Statistics (0.01 % utilization)
  Pkt/Second: 1 (59.5 % multicast)
  Byte/Second: 93 (748 bps)
  Err/Minute: 0 (0 errors)
  Disc/Minute: 0 (0 discards)
  Percent Err: 0.0 % (42 pkts w/o error)
```

Last updated Jun 23, 12:21:02; sample: 37.94 seconds.

Figure 6-9: Typical Network Information.

The Charts Web Page

The Charts web page shows a list of the strip charts associated with a particular map. Clicking on one of the chart names will display the strip chart.

Chart List for "Internal Net"

- [% Errors/Util of DSL circuit](#)
- [Bytes in/out on DSL circuit](#)

Figure 6-10: A typical Charts page. It lists the strip charts for the particular map.

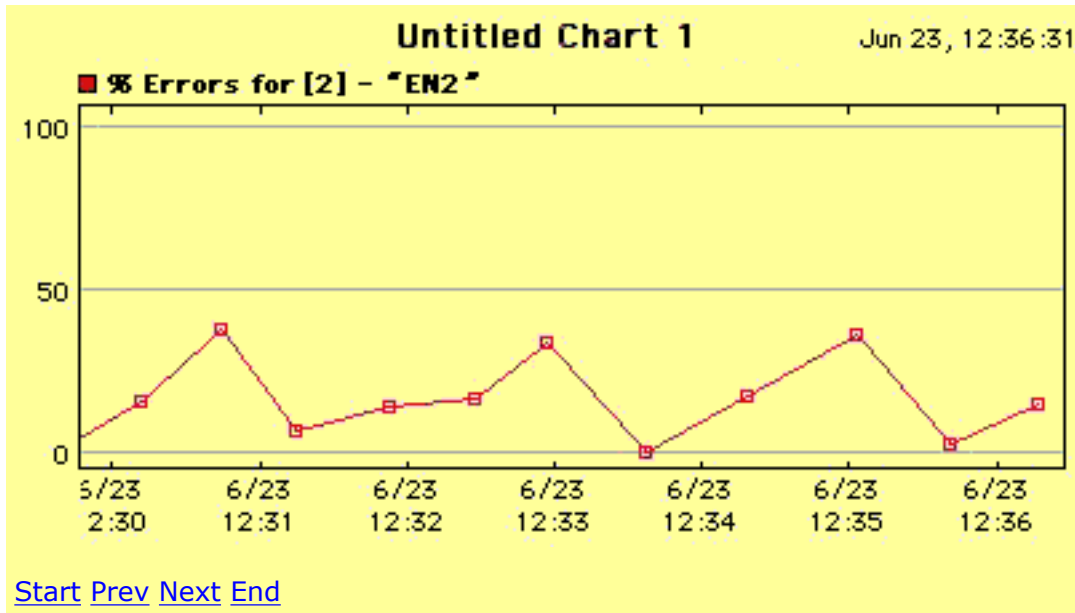


Figure 6-11: A typical strip chart. Clicking the Start, Prev, Next or End links show different parts of the history.

Miscellaneous Links

Map List Page

The **Maps** link leads to a page listing all the maps. Clicking on one of the map links goes to a [map display page](#) that shows a graphic of the map. This page also lists the strip charts associated with each page; clicking on one of those links goes to a [chart page](#).

Map List

- [DNS Names](#)
- [Internal Net](#)
 - [% Errors/Util of DSL circuit](#)
 - [Bytes in/out on DSL circuit](#)
- [Search Engines](#)
- [Various Probes](#)

Figure 6-12: The Map List Page. This shows the list of maps available to view, and any charts within a map.

The About Page

InterMapper's About web page shows information about the InterMapper version, and statistics, such as how long it has been running. This information is identical to that shown in the InterMapper program's About box.

Internet Mapping and SNMP Monitoring for the Macintosh.
Version 3.5 (PPC). Released Friday, July 6, 2001.

For the latest news, visit the [InterMapper Web Page](#). For feedback and technical support, send e-mail to InterMapper@dartware.com.

Registered to:
Tom Terrific

Maximum number of monitored devices: Unlimited
Current number of monitored devices: 10

InterMapper Stats

Memory Used: 1332 K of 5238 K total (25.4 %)

InterMapper Running Time: 6 minutes, 45 seconds
MacOS Running Time: 2 hours, 38 minutes
MacOS System Version: 9.2
Open Transport Version: 2.7.7

AppleTalk Address: 0/78 (router 0/0)
IP Address: 192.168.1.4 (mask 0.0.0.0)

Figure 6-13: The About InterMapper web page. It shows information about the registered owner, the number

of devices being monitored, and statistics about the program's operation.

Telnet Link

The **Telnet** link causes *InterMapper* to launch the Telnet application configured in *Internet Config* and then connect to its own machine.

Built-In Probe Reference

InterMapper has two classes of built-in probes:

Packet-oriented probes like "Ping/Echo", "NTP" and "RADIUS" send UDP or AppleTalk packets to the device being tested and await a correctly formatted response. The timeout period for waiting can be configured in the device's Get Info window. If no response is received within the timeout period, InterMapper tries again by sending another request packet. This process is repeated until either a response is received, or the number of requests sent exceeds the "Number of Lost Packets" threshold set for the map (a default of 3). All packet-based probes check the integrity of the response they receive, and some can set the status of the device (e.g. ALARM, WARNING, OKAY) based on the severity of a problem.

TCP-based probes like "HTTP", "SMTP", and "LDAP" and others test the ability of a server to accept a TCP connection on a specific listening port and respond to a scripted interchange. InterMapper first attempts to connect to the specified port at the device's address. If this connection attempt fails, InterMapper will show the device in the DOWN state. If InterMapper successfully connects to the listening port, InterMapper sends protocol-specific commands through the TCP connection to test the server's responses and compare them to expected values. InterMapper will alter the status of the device (e.g. ALARM, WARNING, OKAY, DOWN) if an error condition is detected, or if the execution of InterMapper's probe is interrupted for any reason. If InterMapper doesn't receive a proper response for 60 seconds, or the TCP connection is lost while waiting for a response, the InterMapper probe will set status of the device to the proper condition.

The definitive description of each probe is contained in the text of the Probe Configuration Window. The descriptions below give an overview of the probe's actions.

Non-Polling

When set to "Non-Polling", InterMapper will not poll the device. Its icon becomes gray, and InterMapper's map no longer reflects the device's state.

Automatic

InterMapper first tries to use the Simple Network Management Protocol (SNMP) to communicate with the device. If SNMP is enabled and a SNMP response is received, the probe type will automatically change to "SNMP" and InterMapper will continue to poll the device using SNMP.

If InterMapper is unable to elicit a SNMP response after a number of attempts, the probe type will automatically change to "Ping/Echo" and InterMapper will continue to test for basic connectivity.

Ping/Echo

For IP devices, this probe sends an ICMP Echo Request to the device's address. For AppleTalk devices, InterMapper sends an AppleTalk Echo Protocol request to the device's AppleTalk address.

The data area in Echo packets is 20 bytes long and contains the version of InterMapper that sent the request.

BlitzWatch

InterMapper sends BlitzMail status queries to a BlitzMail server, and checks for appropriate responses. BlitzMail is a high performance, cross platform client-server e-mail system designed at Dartmouth College. More information is available at <http://www.dartmouth.edu/netsoftware/>.

DHCP

InterMapper sends a DHCP Discover or DHCP Inform request to the device as specified in the configuration

window, and waits for an appropriate response. InterMapper will send the request to the specified BootP Relay Address (normally the IP broadcast address, 255.255.255.255). If the BootP Relay Address is empty, InterMapper will send the request directly to the device. InterMapper will check the response to ensure that the returned IP address is correct, and if specified, that the returned subnet mask and router address are correct.

Domain Name (DNS)

InterMapper sends a DNS "A" record query to the device for the specified Domain Name, and expects to receive the specified IP address in the DNS response. Set *Recursion Desired* to True to allow the DNS server being tested to pass the request to other DNS servers recursively; set it False to force the DNS server being tested to use its own database, and return an error if the address is not available. Use the *Failure Status* pop-up menu to indicate the severity of a failure to receive a response or an incorrect answer in the response.

KeyServer Status

InterMapper sends a KeyServer Status request to the device, and waits for an appropriate response. The format of this packet available from Sassafras Software, <http://www.sassafras.com>

Multicast Listener

InterMapper listens to the stream of packets arriving on the specified multicast address and UDP port. If a packet fails to arrive within the specified time period, InterMapper sets the device status to down. If the "Verify source address" flag is true, then InterMapper will only accept packets from the device's IP address. Packets from other sources will be ignored.

This is most useful for monitoring RTP streams, such as those sent by QuickTime and RealNetworks servers. The settings for this probe are contained in the Session Description Protocol (SDP) file associated with the multicast broadcast. Open the SDP file using a text editor, and locate the required values into the configuration window. Here is an excerpt from an SDP file.

```
o=- 3151391188 3157116984 IN IP4 192.168.152.11      <-- source is 192.168.152.11
s=Sample QuickTime SDP File
i=Copyright @ 2000
a=x-qt-text-nam: My QuickTime broadcast
a=x-qt-text-cpy: Copyright @ My Broadcast
a=x-qt-text-cmt: Broadcast by Sorenson Broadcaster 1.00.307.100
c=IN IP4 224.2.166.240/15/1      <-- multicast address is 224.2.166.240
t=3157116984 0
m=audio 18562 RTP/AVP 12      <-- port for the audio is 18562
a=rtpmap:99 X-QT/600
a=rtpmap:102 X-QDM
a=x-bufferdelay:8
```

Note that an SDP file may describe a broadcast with many tracks (audio, video, text, etc.) Each of these will have a "m= xxxxx port#" line. InterMapper may monitor multiple tracks, but will use a separate device for each.

NTP

This probe sends a client-mode request to an NTP server asking for the current time, and displays an error if no response is returned. This probe does not validate the time in the NTP response.

RADIUS

This probe tests a RADIUS server by sending an Access-Request packet to authenticate a specific user name and password. A RADIUS server will not answer access-requests from a client it doesn't recognize. Before you can use this probe with a particular RADIUS server, you must add the InterMapper computer's IP address to the RADIUS server and choose a "shared secret" for it. The "shared secret" is used by the RADIUS protocol to encrypt passwords in RADIUS requests.

RTMP

InterMapper sends an AppleTalk RTMP RDR Request query of type 3, and waits for a RTMP response.

SNMP

InterMapper queries an SNMP-speaking device to determine the statistics of its interfaces. Each interface is queried for these variables:

ifAdminStatus 1.3.6.1.2.1.2.2.1.7
ifOperStatus 1.3.6.1.2.1.2.2.1.8

InterMapper also sends queries for these variables to detect error conditions:

ifInDiscards 1.3.6.1.2.1.2.2.1.13
ifInErrors 1.3.6.1.2.1.2.2.1.14
ifOutDiscards 1.3.6.1.2.1.2.2.1.19
ifOutErrors 1.3.6.1.2.1.2.2.1.20

Custom SNMP Probes

Custom SNMP probes appear in this position of the popup menu. These probes are discussed in detail in the [Building Custom SNMP Probes](#) chapter. The following SNMP probes can be used as the basis of new user-created custom probes:

Basic OID

InterMapper queries the specified OID and displays its value in the device's popup window.

Cisco

InterMapper queries the CPU utilization and free memory of a Cisco router and places the device in warning or alarm based on user-settable thresholds.

TCP Check

InterMapper queries the MIB-II TCP table to retrieve the number of TCP connections in place. If that count exceeds the parameter, the device is placed in alarm.

TCP-based Probes

4D Server

InterMapper connects to a 4D Server on port 19813, and examines the response for the specified database name.

AppleShare IP

InterMapper sends an AppleShare "Get Server Info" request through a TCP connection to the specified port (default is port 548) on the device being tested. The device is shown as down if the connection cannot complete or if an error response returns.

Note: AppleShareIP versions earlier than 6.3 may have a bug where the AppleShareIP server will hang if it receives an InterMapper *Get Server Info* probe during startup. Version 6.3 and newer do not seem to have this problem.

Basic TCP

InterMapper attempts to establish a TCP connection to the specified port. The device is shown as UP if the connection completes within the default time interval (at which point the connection is disconnected); otherwise, the device is shown as down.

Custom TCP

InterMapper attempts to establish a TCP connection to the specified port. Once the connection has been established, InterMapper sends the "string to send" and waits the interval specified in the "seconds to wait" for a response to return from the device. The response text is examined for the presence of the UP, WARNING, ALARM, or DOWN strings, and the device is displayed in the corresponding state. The device is also considered to be down if none of the response strings are present.

CVS Server

InterMapper attempts to establish a TCP connection to port 2401 on a CVS server. It then attempts to log into the server using the supplied name and password. InterMapper scrambles the password as described in the CVS protocol specification, http://www.loria.fr/~molli/cvs/doc/cvsclient_4.html#SEC4.

DND Protocol

InterMapper connects to a DND (Dartmouth Name Directory) server on port 902 and issues a request for the individual named in the parameter.

FileMaker Pro

InterMapper connects to a FileMaker PRO server on the default port 5003, and returns OK if the connection is successful.

FirstClass Server

InterMapper connects to a port (default is 510) on a FirstClass e-mail server. It sends two CR's, and examines the resulting response for the presence of the specified banner string.

FTP (Login)

InterMapper connects to a port (default is 21) on a File Transfer Protocol server (RFC 959) and attempts to log in using the specified username and password. It then issues a NOOP command, and finally a QUIT command.

FTP (No login)

InterMapper connects to a port (default is 21) on an FTP server (RFC 959). It then issues a NOOP command and a QUIT command. This command is useful for checking an FTP server frequently without filling its log files with login attempts.

Gopher

InterMapper connects to a port (default is 70) on a Gopher server (RFC 1436). It then sends the specified selector string (the default is ""). The presence of any response indicates that the server is up; no error checking is performed on that response.

HTTP

InterMapper connects to a port (default is 80) on an HTTP server (RFC 2068 & 1945). It then sends an HTTP GET request with the specified URL Path (the default is "/"). It then examines the resulting response for the

presence of the "string to verify". You may also specify a name and password to be sent along with the HTTP request to test the authentication.

To test virtual hosting on a particular server, specify a DNS name in the "Host Name" parameter. InterMapper will include that in the HTTP Get request header.

HTTP (POST)

In order to test web form submissions, InterMapper can send an HTTP POST request to the server. InterMapper connects to a port (default is 80) and sends the specified URL Path and Form Data strings in the request. It then searches the response for the "string to verify".

HTTP (Proxy)

InterMapper can test proxy web servers to ensure that they are delivering pages properly. InterMapper sends an HTTP GET request containing a URL of some other host to the proxy server. If the response contains the "string to verify", the proxy server is shown as UP. You may optionally specify a proxy user ID and password. The default for each is empty.

HTTP (Redirect)

InterMapper uses the HTTP (Authenticate) probe to connect to the device on the specified port. It checks the response for the presence of a 301 Moved Permanently status giving the specified "Redirect path".

HTTPS

InterMapper establishes a SSL connection to a HTTPS server (default port is 443), requests a page, and checks for the presence of a specified string on the resulting page. There are parameters to specify the host name, the file's path, the expected string, and an optional user name and password.

Note: With the Classic MacOS (9.2 and earlier) InterMapper uses the built-in URL Access feature to query the HTTPS server. This works fine as long as the server uses certificates derived from well-known Certificate Authorities. If your HTTPS server uses a private certificate, InterMapper will show the server in alarm with a "-6986" error. We are working on this. InterMapper running on MacOS X is not subject to this problem.

HTTPS (Post)

InterMapper establishes a SSL connection to a HTTPS server (default port is 443) and posts form data to the server.

IMAP4

InterMapper connects to a port (default is 143) on an Internet Message Access Protocol - Version 4 (RFC 2060) server. It then issues the IMAP4 CAPABILITY, NOOP, and LOGOUT commands sequentially. It also checks the response of the CAPABILITY command to ensure that the device supports IMAP4 or IMAP4rev1.

IRC

InterMapper connects to a port (default is 6667) on an Internet Relay Chat (RFC 1459) server. It then logs in using the IRC protocol with the specified Nickname, Password, and Username. It then examines the response for the presence of the "string to verify".

LDAP

InterMapper connects to a port (default is 389) on a Lightweight Directory Access Protocol (RFC 2251) server. It then attempts a lookup using the specified "Bind Name" and "Name to Lookup". If no response is returned, or if the response indicates an error, InterMapper shows the device as DOWN.

LPR

InterMapper connects to a port (default is 515) and requests information about a specified queue. If the response is correct, the device is considered up.

NNTP

InterMapper connects to a port (default is 119) on an Network News Transfer Protocol (RFC 977) server and requests information about the specified group with a GROUP command. It then sends a QUIT command to terminate the session.

POP3

InterMapper connects to a port (default is 110) on a Post Office Protocol - version 3 (RFC 1939) server, and verifies that the response contains "+OK".

RTSP

InterMapper connects to a port (default is 554) to on a Real Time Streaming Protocol server (RFC's 1889 & 2326) and verifies its operation. It causes the specified movie to play for the specified number of seconds by sending a sequence of CONNECT, START, and STOP commands.

SMTP

InterMapper connects to a port (default is 25) on a Simple Mail Transfer Protocol (RFC 821) server and attempts to verify an e-mail address. It connects, then sends a HELO command, a VRFY command with the specified "E-mail Address", and then a QUIT command.

Telnet

InterMapper connects to a port (default is 23) on a Telnet (RFC 845) server and attempts log in using the specified name and password.

Customizing Probes

For many Internet services, simply "pinging" a device is not a sufficient test of whether it is operating correctly. Furthermore, InterMapper's built-in probes may not test the kinds of devices you want to monitor.

You can define your own probes. Most of InterMapper's probes are defined by *probe files*, which are text files, and can be duplicated and modified using any standard text editing utility. New probe files become first-class probes, and appear in the **Probe Type**: pop-up menu along with the built-in ones.

It's fairly straightforward to modify the existing files to produce new probes. If you make a new probe file that might be useful, please consider sending it to us at support@dartware.com so that we can make it available to other customers. You can also check the list of probes that have already been contributed at <http://www.intermapper.com/contrib/>.

All InterMapper's probes follow the same logic:

- The probe sends one or more pieces of data, as UDP or AppleTalk datagrams or over a TCP connection to the device being tested.
- The device responds (or fails to respond).
- If there is no response, InterMapper sets the device's status to DOWN.
- InterMapper examines the response(s) from the device, and sets the device's status accordingly.

Probe files are defined by the following attributes. You might want to open a separate window with the [example probe file](#) while reading the subsequent sections.

[Probe File Names](#)

This section defines where probe files are saved on the hard drive, and how the files are named.

[Header Information](#)

This section of a probe file describes how the probe is identified, how its name appears in the **Probe Type**: pop-up menu, and the version numbering system.

[Text Description](#)

This is the text description of the probe that will appear in the user's probe editing window. The text typically describes the parameters to the probe (if there are any), and tells why the probe might be used. There is a markup language that allows a certain amount of formatting and text styling is also described here.

[Parameters to the Probe File](#)

This section defines the parameters for the probe. InterMapper automatically creates fields for these entries when it opens the configuration window.

[Scripting Language](#)

The script is a sequence of statements that govern what data to send out a TCP connection, what response is expected, and the return status to display upon receipt of certain data from the device being probed.

[Comments](#)

Comments may be interspersed anywhere on the probe file. Their format is similar to HTML comments: read the section for the full details.

[Custom SNMP Probes](#)

Custom SNMP Probes give you the ability to query specific MIB variables besides those built into InterMapper. You may also specify thresholds to compare these variables to for warnings and alarms.

Note: Before modifications will become effective, you must choose **Reload probes...** from the **Probe Type**: popup menu in a device's **Get Info** window.

Probe File Names

Probes files are saved in the **Probes** folder of the **InterMapper Settings** folder.

The probe files are named with two parts separated by a period ("."). The parts are:

- a *package* name which must be unique for each organization creating probe files. By convention, the package is composed of the organization's DNS domain name, with the segments reversed. Thus, the built-in probes for InterMapper all have a package of "com.dartware." Other organizations may create and share their own probes, since the file names will not collide. (There is a 31 character limit to Macintosh file names which must be observed.)
- a *probe name* which identifies the probe.

For example, the built-in Custom TCP probe is defined in the file named

```
com.dartware.tcp.custom
```

The package is `com.dartware` and the probe name is `tcp.custom`.

Probe File Header

The *header* of a probe file contains a formal description of the probe. The description is composed of several parts. Each has a name, and a corresponding value, written in this format:

```
part-name = "value"
```

where *value* is enclosed in double-quotes. The parts of a header are:

- **type:** describes the type of the probe file. InterMapper supports "builtin", "tcp-script" and "custom-snmp" probe types. Custom TCP probes should be of type "tcp-script"; custom SNMP probes are of type "custom-snmp".
- **package:** is part of the probe's full identifier. It typically is the domain name of the organization that created the probe, with the labels reversed. For example, all probes created by Dartware, LLC will have packages of `com.dartware`. This guarantees that different organizations may make probes without concern that their probe identifiers will conflict.
- **probe_name:** is the second part of the probe's full identifier. The `probe_name` may be whatever string the creating organization chooses.
- **human name:** is a string that will be displayed in the **Probe Type:** popup menu. This is the way a user may select the probe for a particular device.
- **version:** provides a means for determining which probe file is the current one. The format of the version is "#.#".
- **address_type:** is a comma-separated list of one or more address types. InterMapper implements "IP" and "AT".
- **port_number:** is the IP port used by this probe.
- **old_protocol:** and
- **old_script:** are both used for backward compatibility with existing probes in InterMapper. Newly-created probes do not need the "old_script" part.

Note: The combination of `package.probe_name` together form the probe's full identifier. By convention, this is the same as the filename that holds the probe's information. This is not required, but it's a good idea.

Sample Header Section

This is a sample header from the Custom TCP script.

```
<header>
type = "tcp-script"
package = "com.dartware"
probe_name = "tcp.custom"
human_name = "Custom TCP"
version = "0.1"
address_type = "IP"
port_number = "23"

old_protocol = "8" # Backward compat. with old numbering scheme.
old_script = "8001"
</header>
```

Probe File Description

The Description section of a probe file contains text that will be displayed as a description of the probe in the Probe Configuration window. A sample is shown in Figure B-1. Note that the description can be marked up with bold, italic, plain text. The underlined text becomes a link to an external URL, as described below.



Figure B-1: The Probe Configuration window, showing the description field. Note that the blue underlined links are actually links to the relevant RFC specifications.

The description section is delimited by `<description>` and `</description>` tags. The entire text between those tags is placed into the probe configuration window.

The Markup Commands

The description section can have styled text using markup commands delimited by `<` and `>` characters (Option-`\` and Option-Shift-`\` on the Macintosh keyboard). There may be many markup commands within a single pair of `...<` characters. The [Example Probe File](#) shows a sample description section.

Markup commands apply to all the text that follows it. Subsequent markup commands may add to, or counteract a previous set of markups to the text. The markup commands are:

Command Action

- M** Set the font to Monaco (a monospaced font).
- G** Set the font to Geneva (a proportional spaced font).
- +** Increase the font size by one. Multiples (e.g. "++") will increase by the corresponding amount.
- Decrease the font size by one. Multiples are allowed.
- B** Set subsequent text in bold.
- I** Set subsequent text in italic.
- P** Set subsequent text to plain. This undoes all other stylings
- U** Set subsequent text to underlined. See **Creating a link** section below for making hyperlinks.
- digit** Set text color to one of the following: 0: black; 1: red; 2: blue; 3: gray; 4: light blue; 5: green.

Examples

The following description text would be rendered as shown on the right:

```
«b»Bold text«p» : «i»Italic text«p» Bold text : Italic text
«M l+++»Big red monospace«p» Big red monospace
«2U»http://www.apple.com«P0» http://www.apple.com
```

Creating a link

The last example above shows how to create a link. The "U" markup command indicates that the text between the opening `U` tag and the closing `P` tag should be underlined and displayed in the specified color. Thus, the example above shows `2U` to force the color to underlined blue, and closes with `P0` to set the text back to plain black text.

As a special case, if the only text between the opening and closing tags is a URL (e.g., `http://www.company.net`), InterMapper will treat it as a link and will invoke the user's web browser to retrieve that URL when it's clicked.

Parameters

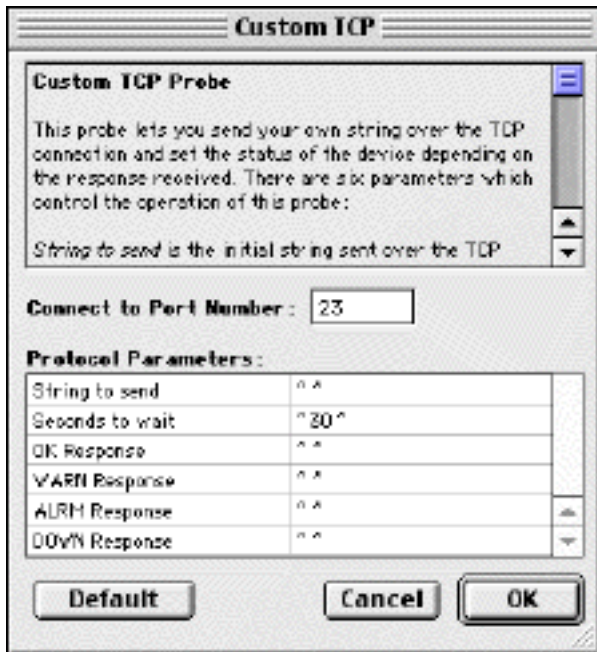
Probes may have parameters which are set in the Probe Configuration window (shown below). These are used for numeric thresholds or strings to be sent or received from the device. The *parameter* section of the defines a set of name/value pairs with the format:

```
"parameter name" = "parameter value"
```

Each parameter name/value will be placed in its own entry field in the Probe Configuration window.

Sample Parameter Section

This is a sample parameter section from the Custom TCP script. Note that the list of parameters at the right matches the values shown in Figure B-2 below.



```
<parameters>
```

```
"String to send" = ""  
"Seconds to wait" = "30"  
"OK Response" = ""  
"WARN Response" = ""  
"ALRM Response" = ""  
"DOWN Response" = ""
```

```
</parameters>
```

Figure B-2: Probe Parameters.

Probe Script Language

InterMapper probe files have a scripting language that can be used to create custom probes. The script statements can send data to the device being tested, examine responses from that device, and return a status based on the response.

All probes have a common flow. They send data (as datagrams or over a TCP connection) to the device to be tested, then examine the response(s). Based on the response, the probe sets the device status (to UP, DOWN, WARN, or ALARM) and a condition string that is a text description of the state.

Script Commands

All script command keywords are 4 letters long, and they are case-sensitive. All commands MUST be in UPPER CASE. There must be white space between the commands and each argument. You can include other text (e.g. comments) after the first argument, as long as it is separated by white space from the remaining arguments.

Example: The command statement `MTCH "blah" else goto #7` is treated exactly the same as `MTCH "blah" #7`. When parsing the statement, InterMapper ignores the "else goto" part. This allows you to include comments to make the behavior of the script more obvious. This extraneous text does not have to be in uppercase.

String Arguments ("string")

Some commands take string arguments. String arguments must be in double-quotes. The following special characters may be included by using a backslash escape code:

- `\r` Carriage Return
- `\n` Unix Linefeed
- `\t` Horizontal Tab
- `\f` Formfeed
- `\b` Backspace
- `\v` Vertical Tab
- `\a` Alert (bell) Character
- `\"` Double Quote
- `\\` Backslash
- `\ooo` Octal Number
- `\xhh` Hexadecimal Number

Example: `"\tThis sentence is preceded by a tab, and followed by a carriage return and linefeed.\r\n"`

String Matching

Some commands specify a string to match. By default, string-matching is case-sensitive. Place an 'i' after the final quote if you want the matching to be case-insensitive.

Example: `"fred"` matches only `"fred"`.

Example: `"fred"i` matches `"fred"`, `"FRED"`, or `"FrEd"`.

In some cases, it is convenient to match a more general pattern. InterMapper lets you define an expression containing special characters which match any character. When an 'r' follows the final quote, a '.' in the expression matches any single character.

Example: `".red"r` matches `"fred"`, `"Fred"`, `"tred"`, `"bred"`, etc. It does not match `"freD"`.

You can combine the 'r' and the 'i' to indicate both expression matching and case-insensitivity.

Example: `".red"ri` matches `"fred"`, `"freD"`, `trEd"`, etc.

Note: Previous versions of InterMapper allowed a perl-like syntax of `m"xxx"` to search for a pattern that might include a `.`. This continues to work in version 3.6, but is deprecated, and no longer recommended for new scripts.

Numeric Arguments

Some commands take numeric arguments. Numeric arguments are formed by a `#` sign followed by digits.

Example: `WAIT #30`

In many cases, numeric arguments are used to indicate the script statement number to go to when a failure occurs. Since it is convenient to express these jumps as relative offsets, there is a special notation for this. Include a sign (`'+' or '-'`) after the `#` to express a relative offset from the current statement.

Example: `GOTO #+2`

If a command takes a numeric argument, but you do not include it, the default value is 0. If you specify 0 as the statement to go to when the script fails, this will terminate the script with a `DOWN` condition.

Labels and Transfer of Control

Labels serve as places to receive control from elsewhere in the script. Labels have the form shown below, and must be alone on a line:

```
@label_name
```

For example, a label could be defined as `"@IDLE"`.

Control may be transferred by inserting the label definition in the statement. An example is:

```
WAIT #30 seconds else goto @IDLE
```

Control may also be transferred by relative amounts. The form for these relative transfers is `"#+n"` or `"#-n"` to transfer forward or backward `n` statements (respectively). An example relative transfer is

```
MTCH "${WARN Response}" else #+2
```

Variables

Before each script statement is processed, it may have certain variables substituted into it. These variables names and their default values are defined in the `<parameter>` section of the probe file.

Variable names appear enclosed by curly braces following a dollar sign (`$`). They are case-insensitive.

Example: `${Password}` and `$(password)` are treated as the same variable

Built-in Variables and Macros

InterMapper has a number of built-in variables. To prevent confusion with user-defined variables, all built-in variables are prefixed with a single underscore.

`$_REMOTEADDRESS` The IP address of the other side of the connection.

`$_REMOTEPORT` The TCP port of the other side of the connection.

`$_LOCALADDRESS` The IP address of this side of the connection.

`$_LOCALPORT` The TCP port of this side of the connection (usually an ephemeral port).

`$_GMTTIME` The current GMT time in RFC 822 format.

`$_VERSION` The current version of InterMapper (e.g. "3.5b4").

`$_IDLETIMEOUT` The number of seconds specified as an IDLE timeout.

`$_STRINGTOMATCH` The last argument to a `MTCH` or `EXPT` command (to be used for error messages.)

`$_IDLELINE` The line number in the TCP script where we were IDLE.

`$_SECSCONNECTED` The number of seconds the TCP probe has been connected.

Macros are expressions that modify a particular input string to produce another string. The built-in macros are:

`${_LINE:<line>}` The first `<num>` characters of the last line received.

`${_BASE64:<param>}` The Base-64 encoding of the string that follows the ":".

`${_CVSPASSWORD:<param>}` The IP address of the other side of the connection.

Script Failure

Certain script commands may fail, either because they are malformed or because an unexpected situation occurred. For example, the script could transfer to a non-existent command, it could fail to match a string it expects, or an unexpected disconnection could occur. In each case, the script will immediately branch to a failure handler in the script. Each command that can fail takes the statement number of the failure statement as a numeric argument. If this number is omitted, the script will terminate with a "DOWN" status.

In the following example, the MTCH command succeeds if the incoming line of data contains "220". If the command fails, the script branches to statement 3: MTCH "220" ELSE #3

(If the script is idle for too long, it may go to a special "idle" handler. See the WAIT command for more details.)

Command Reference

This is a list of each command defined in the InterMapper Scripting Language.

DONE status ["message"]

This command terminates the script with the specified condition, which must be one of [OKAY | WARN | ALRM | DOWN]. The optional "message" parameter lets you provide more detail about the condition. The status values for the DONE command must be in UPPERCASE.

Example: DONE ALRM "[HTTP] 500 Response received." will set the status of the device to "Alarm". The condition of the device (which is displayed in the popup window and the Device List window) will be set to "[HTTP] 500 Response received." to give the user an indication of the reason for the alarm.

Possible Failures: This command can't fail.

Tip: If the final statement of your script is not a DONE command, the script automatically terminates with a "DONE OKAY" status.

STAT status ["message"]

The STAT command lets you specify the status (which must be one of [OKAY | WARN | ALRM | DOWN]) of the device when the script ends and optionally, the condition string for the device, but it doesn't terminate the script. A subsequent STAT or DONE command will override the value set by this command.

EXIT

The EXIT command terminates the script, setting the status and condition string to whatever is specified by a previous STAT command.

SEND "string"

Sends the specified data to the remote device. If you want to send a line of data, you must explicitly specify the CR-LF using the quoting convention.

Example: SEND "Greetings!\r\n" transmits the data "Greetings!" followed by a CR-LF over the TCP connection to the server.

Possible Failures: This command can't fail. If the data can't be sent because of a network failure or device failure, the failure will show up in a subsequent EXPT or MTCH command.

WAIT #secs #idlefail #discfail

Set the timeout to wait for subsequent string matching commands. The first parameter is the time to wait for

a matching string. If a script doesn't specify the timeout, the default timeout is 60 seconds.

If a second parameter is present, the script will jump to this line number when the probe is idle for the specified number of seconds. This idle handler supercedes the error line number specified by the EXPT, SKIP, or MTCH commands. If the #idlefail parameter is not included, the script will branch to the failure handler of the current command.

If a third parameter is present, the script will jump to this line if the probe is unexpectedly disconnected. This allows you to identify scripts that fail because of a TCP disconnection.

Possible Failures: None

Tip: You should specify all three parameters in the WAIT command.

EXPT "string" #fail

Search for the specified string in any number of incoming lines. (The mnemonic is "EXPeCT".) If the string is found, the script falls through to the next statement, otherwise it will transfer to the #fail statement.

Possible Failures: This command can fail if the expected text is not received before the connection closes. In that event, the script will jump to the statement specified by #fail. However, if the timeout specified by a previous WAIT command expires before the connection closes, the script will jump to the #idlefail line specified by the WAIT command instead.

MTCH "string" #fail

Search for the specified string in the next incoming line. If found, the script falls through to the next statement.

Possible Failures: If the next line does not contain the desired string or the connection closes before the next line can be read, this script will fail. In either case, the script will jump to the statement specified by #fail. If the idle timeout expires instead, the script will jump to the #idlefail line specified by the previous WAIT command.

NBGT #arg1 #arg2 #line

Numeric Branch Greater Than command. If *arg1* is greater than *arg2*, the script jumps to #line.

Note: Use the leading # to force InterMapper to treat the arguments as numeric values.

SKIP "string" #fail

Skip all incoming lines containing the specified string. The script falls through to the next statement when an incoming line does not contain the "string".

Possible Failures: This command can fail if the connection closes. If the WAIT timeout expires, the script jumps to #idlefail.

DISC #discfail

Set the line number of the script to jump to if the probe is suddenly disconnected. This allows you to identify scripts that fail because of a TCP disconnection. The script's disconnect line can also be set using the third parameter to the WAIT command.

CONN #timeout ["TELNET"]

The CONN command specifies the connect timeout of the probe and whether to process Telnet options. This command must be the first statement of the script. When the script executes, the parameters of the CONN statement determine the options on the connection to the remote computer.

The first parameter is the number of seconds to wait while connecting before giving up.

If there is a second parameter and it is "TELNET" (including the quotes), then the connection is created in a mode where the TCP stream will automatically process and negatively acknowledge any incoming Telnet options. This allows a Telnet probe to ignore the telnet options and work in simple line-by-line mode for the remainder of the script.

Possible Failures: None

PORT #port_num #connect_timeout

Deprecated This command is no longer required in a script because the remote port number is now a separate parameter in the configuration dialog.

If present, this command must be in the first statement of the script. The first parameter specifies the default TCP port to connect to on the remote computer. The #connect_timeout parameter is the number of seconds to wait for the probe to connect.

Possible Failures: None

GOTO #statement

Immediately branches to the specified statement number.

Possible Failures: If the statement number is out of bounds, the script terminates with "DONE DOWN".

LINE [ON | OFF]

Specifies whether the script should read incoming data as lines or raw data. By default, the script reads in line mode. That is, each line is terminated by a CR-LF or just a plain LF. If you issue the LINE OFF command, data is read without regard for line delimiters.

Reading raw data is useful for scanning HTTP data since web pages are not necessarily broken into lines. InterMapper's TCP probe has a maximum line buffer of 500 chars, so if lines are longer than that, they may be treated as separate lines.

Tip: Once you've matched some data in line mode, you shouldn't match any more because your position in the buffer is not restored and you may miss something.

Possible Failures: None

CHCK "string" #fail

Checks if "string" is non-empty. If the string is empty, the script jumps to the specified #fail line. This command can be used to construct scripts whose control changes depending on whether an optional parameter is supplied (ie like an "if" statement).

Possible failures: None

Annotated Example of the FTP (Login) Script

```
01) PORT #21 (default tcp port)
02) WAIT #30 seconds
03) EXPT "220 " else goto -1- #14
04) SEND "USER ${User ID}\r\n"
05) MTCH "331" else goto -2- #16
06) SEND "PASS ${Password}\r\n"
07) MTCH "230" else goto -3- #20
08) SEND "NOOP\r\n"
09) MTCH "200" else goto -4- #24
10) SEND "QUIT\r\n"
11) EXPT "221" #+1 (i.e. can't fail)
12) DONE OKAY
13)
14) DONE DOWN "[FTP] Unexpected greeting from port ${_REMOTEPORT}. ${_LINE:50})" -1-
15)
16) MTCH "500" else goto #+2 -2-
17) DONE ALRM "[FTP] Port ${_REMOTEPORT} did not recognize the 'USER' command."
18) DONE ALRM "[FTP] Unexpected response to USER command. (${_LINE:50})"
19)
20) MTCH "530" else goto #+2 -3-
21) DONE WARN "[FTP] Incorrect login for \"${User ID}\"."
22) DONE ALRM "[FTP] Unexpected response to PASS command. (${_LINE:50})"
23)
24) DONE ALRM "[FTP] Unexpected response to NOOP command. (${_LINE:50})"
```

Explanation of the Script

```
01) PORT #21 (default tcp port)
02) WAIT #30 seconds
```

The PORT command at the beginning of the script specifies the default TCP port number for FTP, port 21. Then the WAIT command specifies that if the script doesn't hear responses back within 30 seconds, it will fail.

```
03) EXPT "220 " else goto -1- #14
```

FTP servers normally send one or more "220" lines to greet new FTP control connections. Our script scans the incoming lines for "220 ". Note the space following the 220; we don't want to match "220-"; the dash indicates there are still more 220 lines to be read -- we only want to match the final 220 line.

If the script fails to find "220 " before the connection closes or within 30 seconds, the script branches to statement 14. The "-1-" is an arbitrary label used to make the destination of the branch more easily visible. The string "else goto -1-" has no function (except readability) in the script command text; this statement could have been written equally well as EXPT "220 " #14 . Note that statement #14 also has comment of "-1-" to show it is the destination.

```
04) SEND "USER ${User ID}\r\n"
```

Next send the FTP USER command. With this command, we send the user ID specified by the user, e.g. "anonymous". Note that you must include the carriage-return and line-feed at the end of the string sent, to denote the line ending.

```
05) MTCH "331" else goto -2- #16
```

The script looks for the 331 response to the USER command. If something else arrives, the script jumps to statement 16. Unlike the EXPT command, the MTCH command fails immediately if the next line doesn't contain the required text.

[...] (Skipping down to statement 16).

```
16) MTCH "500" else goto #+2 -2-
17) DONE ALRM "[FTP] Port ${_REMOTEPORT} did not recognize the \"USER\" command."
18) DONE ALRM "[FTP] Unexpected response to USER command. (${_LINE:50})"
```

We arrive at statement 16 only if statement 5 fails; that is, if we get an unexpected response to the USER command. We test the response to see if it matches "500" which would indicate that the command isn't supported. This is possible if you accidentally try to pass the USER command to a TCP service other than FTP. If the server's response matches "500", we terminate the script with the device in the ALARM status (statement 17). Our message reports that the server did not recognize the USER command.

If the server's response does not match "500", the script skips two lines to statement 18. This statement terminates the script with the ALARM status and uses the `${LINE}` macro to include the first 50 characters of the response line in the message.

Customized Popup Windows

Devices with custom TCP or SNMP probes can override the default popup window that would be shown. This is done using an optional section that defines data that will be added to the bottom of the popup window.

Custom SNMP Probes

Custom SNMP probes have a `<snmp-device-display>` section that describes the text that will be shown in the pop-up window. Probe variables will be replaced with their values in the popup text.

The default font for the popup window's text is Monaco 9 (a monospaced font), so alignment of text is straightforward. You can change the appearance of the text in the popup using the markup commands of the [Description](#) section.

Here is a sample `<snmp-device-display>` section. Note that the variables will be replaced with the values retrieved from the device.

```
<snmp-device-display>
«B5»Custom SNMP Probe-reb«0P»
  «4»ipForwDatagrams:«0» ${ipForwDatagrams} datagrams/sec
  «4»ipInHdrErrors:«0» ${ipInHdrErrors} errors/minute
  «4»tcpCurrEstab:«0» ${tcpCurrEstab} connections
</snmp-device-display>
```

Custom TCP-Based Probes

TCP-based custom probes may have an optional `<script-output>` section in which to display their results. The data in this section will appear in a popup window when you click and hold on the device. The format of this section is the same as the `<snmp-device-display>` section described above.

Comments

Comments in InterMapper probe files are quite similar to those in HTML. The comments may be interspersed anywhere in a probe file.

Note that HTML comments have a complicated syntax that can be simplified by following this rule:

Begin a comment with "`<!--`", end it with "`-->`", and do not use "`--`" within the comment.

Use this rule with InterMapper as well.

Example TCP Probe File

The following is the Dartware-provided probe for the Custom TCP script.

```
<!--
Custom TCP (com.dartware.tcp.custom)
Copyright © 2000 Dartware, LLC. All rights reserved.
-->

<header>
type = "tcp-script"
package = "com.dartware"
probe_name = "tcp.custom"
human_name = "Custom TCP"
version = "0.1"
address_type = "IP"
port_number = "23"

old_protocol = "8" # Backward compat. with old numbering scheme.
old_script = "8001"

</header>

<description>

  GB Custom TCP Probe P

  This probe lets you send your own string over the TCP connection and set the status of the device
  depending on the response received. There are six parameters which control the operation of this probe:

  i String to send p is the initial string sent over the TCP connection. This could be a command which
  indicates what to test, or a combination of a command and a password. The string is sent on its own line,
  terminated by a CR-LF.

  i Seconds to wait p is the number of seconds to wait for a response. If no response is received within
  the specified number of seconds, the device's status is set to DOWN.

  i OK Response p is the substring which should match the device's "ok response". If it matches the first
  line received, the device is reported to have a status of OK.

  i WARN Response p is the substring which should match the device's warning response.

  i ALRM Response p is the substring which should match the device's alarm response.

  i DOWN Response p is the substring which should match the device's down response.

  If InterMapper cannot connect to the specified TCP port, the device's status is set to DOWN.

</description>

<parameters>

"String to send" = ""
"Seconds to wait" = "30"
"OK Response" = ""
"WARN Response" = ""
"ALRM Response" = ""
"DOWN Response" = ""

</parameters>

<script>
```

```
PORT #23 (default tcp port) #60 (connect timeout in secs)
SEND "${String to send}\r\n"
WAIT #${Seconds to wait} else goto @IDLE
EXPT ".r" else goto @DISCONNECT
MTCH "${OK Response}" else #+2
DONE OKAY "[Custom] Response was \"${_LINE:50}\"."
MTCH "${WARN Response}" else #+2
DONE WARN "[Custom] Response was \"${_LINE:50}\"."
MTCH "${ALRM Response}" else #+2
DONE ALRM "[Custom] Response was \"${_LINE:50}\"."
MTCH "${DOWN Response}" else #+2
DONE DOWN "[Custom] Response was \"${_LINE:50}\"."
```

@IDLE:

```
DONE DOWN "[Custom] Did not receive a line of data within ${Seconds to wait} seconds. [Line
${_IDLELINE}]"
```

@DISCONNECT:

```
DONE DOWN "[Custom] Connection disconnected before a full line was received."
```

</script>

InterMapper Custom Probe Builder Application

You can make your own probes to perform tests not provided by InterMapper's built-in probe types. These are called *custom probes*, and they have two types:

- Custom SNMP probes
- Custom TCP probes

The Custom Probe Builder application currently only creates and edits Custom SNMP probes. The remainder of this page describes how to use the Custom Probe Builder application to edit the built-in probes or make your own.

Opening, Creating, and Saving Files

The Custom Probe Builder uses standard file open/new/save conventions. When launched, the application creates a new empty probe window. Choosing **Save...** will save the file to the hard drive. (The default file probe file location is in the *InterMapper Settings:Probes* folder.) You can also open an existing probe file (**Open...**).

Note: It is safe to edit files while you're running InterMapper. InterMapper only examines the newly-edited probe files when you use the **Reload Probes...** item in the device's **Probe Type:** popup menu.

Creating a New Probe

Double-click the Custom Probe Builder application to make a new probe, or choose **New...** from the **File** menu. It will open a window like the one shown at the right.

Each probe name and file name must be unique. To do this, we recommend you use this strategy for filling the three fields:

- **Human-readable name:** This will appear in the [Probe Type popup menu](#).
- **Prefix:** This should be the reversed domain name of your institution, e.g., company.com becomes "com.company"; montgomery.k12.md.us would become "us.md.k12.montgomery".
- **Suffix:** This should be a unique tag for your probe, such as "snmp.wxyz" for a WXYZ probe.

After filling in these fields, click **Create** to create the file. You'll see a window like the one shown below.

Editing an Existing Probe

To edit an existing probe, Open it from the File menu. You will see a window similar to the one below.

Building a New Probe: Basic Information

When creating a new probe, you must create two items: a (human-readable) Probe Name that appears in the Probe Type popup menu and also a name for the file on disk.

Probe Name This is the name that will appear in the Probe Type: popup menu. This may be any text, but by convention, it has "SNMP" at the start of the name.

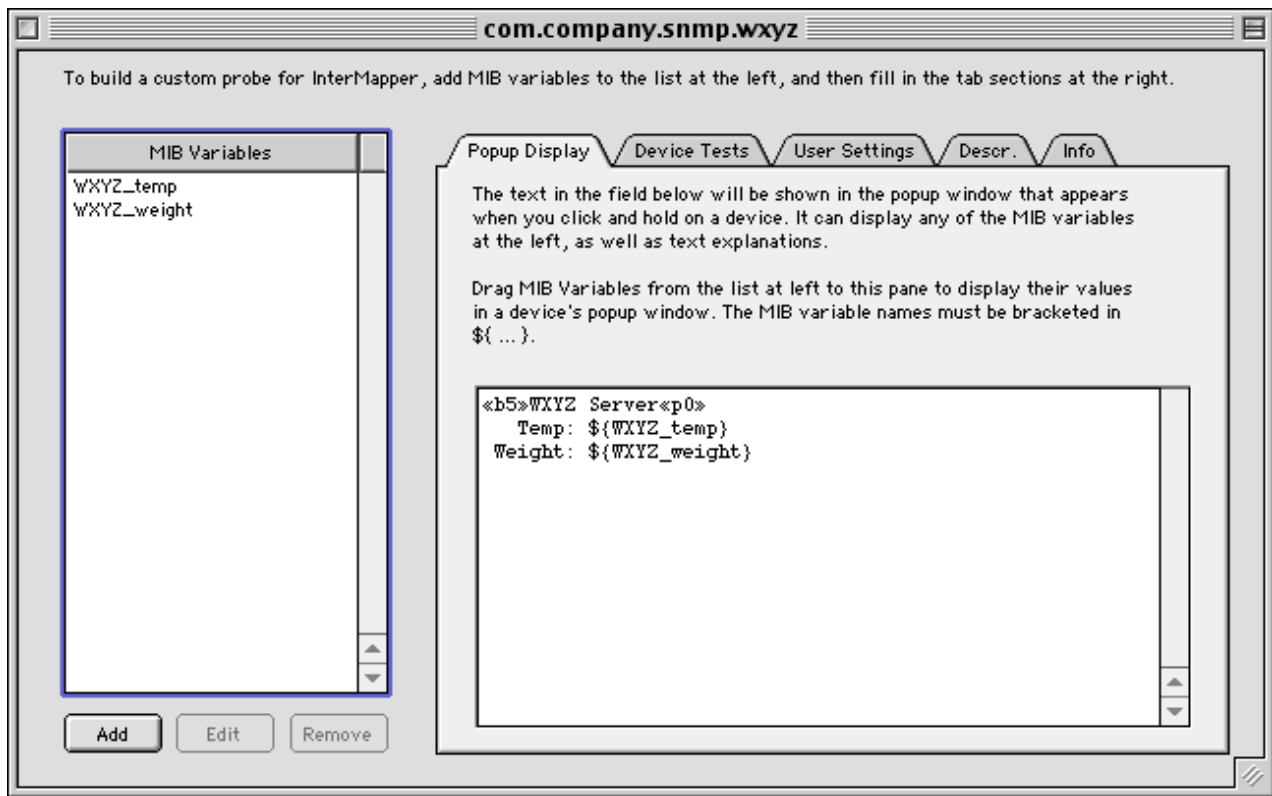
Probe Name (human-readable):

File Name The name of the file (on disk) is composed of two parts, a prefix and a suffix. The prefix may be any string. To keep probes unique, we recommend that you reverse the parts of your domain name to form the prefix of all probes you produce. For example, Dartware's probes are all use "com.dartware" in their names.

Probe Prefix:

Probe Suffix:

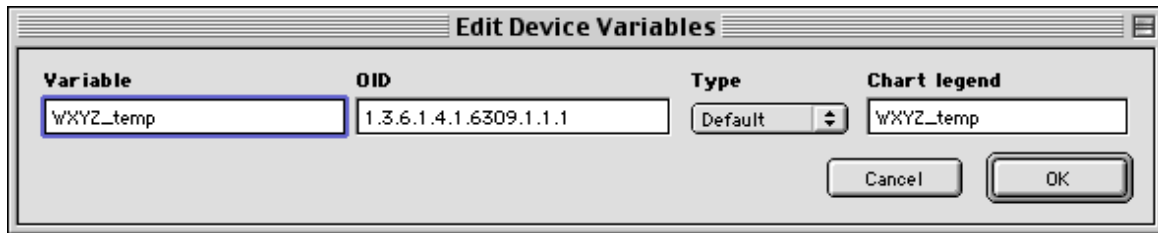
Resulting filename: com.company.snmp.wxyz



This window shows a list of MIB variables at the left, and a series of tabbed panes on the right side of the window. The MIB variables are to be retrieved from a device and tested to determine the devices's status. The tabbed panes allow you to set thresholds, comparison expressions, and condition codes.

Entering MIB Variables

First, you should enter the information about the MIB variables that you wish to monitor. To do this, click the **Add** button below the MIB Variables list. You'll see a window like this:



Fill in the fields: a name for the MIB variable (don't use spaces within the name), the Object ID (OID) for that variable, its format ("Default" is usually acceptable), and an optional legend to be displayed in a strip chart. Click **OK** to add the MIB variable to the list. Add as many MIB variables as you wish to monitor.

The Popup Display Pane

The **Popup Display** pane shown in the figure above define the contents of the popup window. This can contain both static text as well as any MIB or User Setting Variables defined in the probe file. Note that the variable names must be enclosed in "\${...}" characters. InterMapper will substitute the current value for the variable whenever the menu is popped up.

For example, in the figure above, the WXYZ_temp and WXYZ_weight variables will be displayed.

Text in the popup window may be marked up using the commands described on the [Description page](#).

The Device Tests Pane

InterMapper uses the expressions from the **Device Tests** pane (shown at right) to determine the device's status. If the comparison expression is true, then the device goes to the corresponding status, and its condition string is set to the string in the right-most column (if it's present).

For example, the `${WXYZ_weight}` MIB variable will be compared to the `${user_weight_alarm}` value (set in the User Settings pane, described below). If `${WXYZ_weight}` were greater than `${user_weight_alarm}`, then InterMapper would place the device in alarm, and assign it a condition string of "High weight".

InterMapper makes the comparisons in top-to-bottom order, and when it finds an expression that's true, it stops the search and uses the corresponding status and condition string. If no expressions match, the device status is set to OK (green) with an empty condition string. To change the order of the comparisons, select one of the entries in the list and click the **Up** or **Down** buttons to move it to the proper position.

To create new Device Tests, click the **Add** button. You'll see a window like the one below. Fill in the expression to evaluate, select a status from the popup menu, and optionally, enter a condition string to indicate the reason for the alarm/warning/down status. You may use MIB variables, User Settings (described below), numbers or strings in the comparisons.

Device Tests: If the expression below matches, the device will be set to the indicated status, and display the corresponding Condition string.

Expression to evaluate `${WXYZ_weight} > ${user_weight_alarm}`

Status Alarm

Condition String High weight

OK Cancel

User Settings Pane

The **User Settings** are parameters to the probe that the user may control. When you create a setting, you supply a default value, but the user may override that default with their own value.

User settings are variables: when used in expressions, their names must be enclosed in `${...}` just like MIB variables.

For example, in the **Device Tests** pane above, the expression used the `${user_weight_alarm}` variable. This was defined here in the **User Settings** pane. As shown, it has the (default) value of 5 in the figure at the right.

To create or edit a user setting, click the **Add** (or **Edit**) button. You will see an editing window like the figure below. Click the **OK** button to retain the information. You may add any number of User settings to a probe.

Enter the name and the default value of a setting that will be compared to a MIB variable to determine a device's state.

User Setting `user_weight_alarm`

Value 5

OK Cancel

Description Pane

The **Description Pane** gives a text description about the probe. This is free text: it has no effect on the probe's operation. It will be displayed in the top half of the probe configuration window, as shown in [Appendix B](#).

You should consider giving detailed information about the probe here, since this will be the primary source of information about the probe for most users.

The description text is marked up with the commands described on [Appendix B](#). This allows you to vary the color, size, font, and other attributes of the text in the probe's description.

Device Tests specify how InterMapper should set a device's status. The expressions below should compare a MIB variable to a fixed value or one of the User Settings on the next tab. If the expression returns "true" then the device takes on the specified status and condition string.

Status	Expression	Cond. String
Alarm	<code>\${WXYZ_weight} > \${user_weight_alarm}</code>	High weight
Alarm	<code>\${WXYZ_temp} > \${user_temp_alarm}</code>	High temp
Warning	<code>\${WXYZ_weight} > \${user_weight_warning}</code>	
Warning	<code>\${WXYZ_temp} > \${user_temp_warning}</code>	

Up Down Edit Add Remove

User Settings are the alarm and warning limits. InterMapper compares these settings to the MIB variables to determine a device's status. Users may set different thresholds for each device when they set the probe type.

User Setting Name	Value
<code>user_weight_alarm</code>	5
<code>user_weight_warning</code>	2
<code>user_temp_alarm</code>	100
<code>user_temp_warning</code>	90

Edit Add Remove

Info Pane

The **Info Pane** contains miscellaneous information about the probe file. The information includes:

- The file name **Prefix** and **Suffix** will have been set when creating the file, but you may edit them here. Be sure that you save the file with a name made up of these two parts.
- The human-readable **Probe name** is the name that will be shown in the **Probe type**: pop up menu.
- The **Port** is the TCP/IP port that the SNMP queries will be sent to. Default is 161.
- The **Version** is the version number of the file. If two files have the same prefix and suffix, InterMapper chooses the file with the newest version number. If two files have the same prefix, suffix, and version number, InterMapper uses the one with the latest last-modified date.
- The **Address type** specifies whether the probe is to be used with the IP protocol ("IP"), AppleTalk ("AT"), or both.
- Checking the **Use SNMP Get-Requests** box forces InterMapper to issue Get-Requests for the exact OID rather than using a Get-Next-Request for the previous OID. This is useful for devices that do not implement Get-Next-Request.
- Checking the **Use SNMPv2c** box forces InterMapper to issue SNMP v2c format queries. Use this for certain extended MIB variables such as 64-bit counters.
- Checking the **No MIB-II queries** prevents queries on any MIB-II MIB variables such as sysUpTime, which InterMapper normally uses to determine a device's up-time. Use this for devices that only implement a private MIB, not the MIB-II.
- The **Comments** can describe the probe, the author(s), creation and modification dates, and any

Popup Display Device Tests User Settings Descr. Info

Enter text that describes the probe. This text will be visible in the Probe Configuration Window. Use << and >> (Option-\ and Option-Shift-\ respectively) and the commands below to mark up the text.

```
<<b>WXYZ Server Probe<<p>
```

This is a probe file for a fictitious "WXYZ Server". It has MIB variables for temperature and weight, which are described in the MIB Variables at the left.

Each time the device is probed, InterMapper tests the comparisons defined in the <<i>Device Tests<<p> against certain <<i>User Settings<<p>. It then sets the device's status and condition string according to the results of the test.

Popup Display Device Tests User Settings Descr. Info

Information about this probe. Set the file name Prefix and Suffix fields that go to make up the file name, and the Probe name so your probe won't conflict with the built-in probes. The other defaults are usually correct, and can be left alone.

Prefix: Suffix:
e.g. com.company e.g. snmp.xxxxxx

Probe name: Version:

Port: Address type:

Use SNMP Get-Requests only No MIB-II queries
 Use SNMPv2c

Comments: Enter your name, company name, copyright info, date, revision history, etc. here. This text is optional and is only for your information.

```
This is a probe file for the fictitious WXYZ Server. It shows the facilities of the InterMapper Custom Probe Builder program.
```

Probe file created on 28 Dec 2001 reb

revisions to it. Comments will always be ignored, and will not affect the operation of the probe file.

After editing all the panes of this window, save the file to InterMapper's *InterMapper Settings:Probes* folder. Then use InterMapper's Reload Probes command (in the Probe Type popup menu) to force InterMapper to reload the probes. You will see a confirmation window like the one below:



Status of the Custom Probe Builder Application

The current version of the Custom Probe Builder is 1.0d5. It is *test* software: you should always keep backup copies of the files you're working on. You can download the latest version of the program from:

<http://www.intermapper.com/binaries/imcustomprobebuilder.sit>

Custom Probe Builder 1.0d5 - 28 Dec 2001

- [FEATURE] Reads and writes Default, Discovery, Invisible, noMIB2, and SNMPv2c flags in the <header> section
- [FEATURE] Preserves (but ignores) unknown lines of <header> section
- [FEATURE] Custom Probe Builder currently only handles "custom-snmp" probes. It now politely ignores "built-in", "tcp-script", and "url-access" probe files
- [LATERAL] Rearranged the logic for determining which sections are required.
- [BUG FIX] No longer beeps when logging error messages

Custom Probe Builder 1.0d4 - 2 Dec 2001

- [FEATURE] Significant user interface reorganization to show MIB variables on left, with remaining information in tabbed pane on the right of the window
- [FEATURE] It is possible to drag a MIB name to an editable field to copy it. The "\$\{...}" markup will automatically be added if it's proper.
- [FEATURE] Renamed tabs to make their function more recognizable
- [FEATURE] "New Probe..." now opens a new window that requests the minimal information for the probe. (Is this a wizard? I don't think so - I hate wizards...)
- [FEATURE] Always add the probe creation date to the comments field for a new probe.
- [FEATURE] New Log window shows history, files opened, errors discovered.
- [FEATURE] Parses new <snmp-device-properties> section for pdutype and noMIB2 information
- [FEATURE] Checks Get-Request and noMIB2 boxes in the "Info" tab as required.
- [BUG FIX] All editing operations now mark a document dirty.

Custom Probe Builder 1.0d3 - 26 Oct 2001

- [FEATURE] First version good enough to show outside Dartware.
 - [FEATURE] Added "Green" color legend to description & popup window text.
 - [BUG FIX] Editing popup text now marks window as dirty.
-

Custom Probe Builder 1.0d2 - 23 Oct 2001

- [FEATURE] Changed editing dialogs to mark document dirty whenever changes occurred.
 - [BUG FIX] Fixed incorrect text in status popup (was "Warn"; should have been "Warning")
-

Custom Probe Builder 1.0d1 - 5 Jul 2001

- [FEATURE] Initial version. User interface was a single window, with separate tabs for each section of the file.

If you have comments, suggestions, or bug reports, please send them to intermapper@dartware.com.

Errors with Custom Probes

Q: The popup window for a device shows a "Reason: No SNMP Response." at the bottom of the popup window. Why?

A: There are several reasons that InterMapper might not be able to retrieve SNMP information from a device. The most common two are that a) the device doesn't speak SNMP, or b) you haven't entered the proper SNMP read-only community string. The [SNMP FAQ](#) lists many other reasons.

Q: When I build a custom probe, the popup window shows "[N/A]" for certain values. Why?

A: It probably means that there is an error with the OID for one of the device variables. Open the Debug window, and look for entries in this format. (To open the Debug window, hold down Cmd-Option-Shift and press "Z".)

```
12:57:00 router.example.net.: SNMP error status [[query = 28]] noSuchName (2), index = 3
  1) 1.3.6.1.2.1.1.3: NULL
  2) 1.3.6.1.2.1.1.1: NULL
  3) 1.3.6.1.7.1.1.4: NULL
  4) 1.3.6.1.2.1.1.6: NULL
```

Note that the first line above shows a "noSuchName" error for index 3. Look at the following lines to find item 3, and check that OID very carefully. In this example, the proper OID should have a "2" in place of the "7" that's there.

Q: When I build a custom probe, the popup window shows "[noSuchName]" for certain values. Why?

A: It probably means that there is an error with the OID for one of the device variables. Open the Debug window, and look for entries in this format. (To open the Debug window, hold down Cmd-Option-Shift and press "Z".)

```
13:17:59 OID Error: GetNextRequest from 192.168.1.1 expected 1.3.6.1.2.1.2.2.1.2.10; got 1.3.6.1.2.1.2.2.1.3.1
```

In this case, the desired value is from a non-existent table row. (The OID 1.3.6.1.2.1.2.2.1.2 is the ifDescr for an interface on a device. The index (.10) indicates which row to retrieve. But when InterMapper requested that row, it learned it was not present.) Consequently, InterMapper displays the "noSuchName" value.

Measuring Response Times

InterMapper 3.6 introduces the ability to measure the response times of devices being tested. The times are measured in milliseconds.

- **TCP-based probes:** InterMapper can measure both the time to establish the connection and the time for various portions of an interaction. These times can be charted and logged.
- **Timing of Pings and other datagram-based probes:** InterMapper measures the interval between sending the ping and receipt of its response time. The time is displayed in the popup window, and is chartable and loggable.

Time Measurement Probe Variables

TCP Timers are:

Connection initiation interval: `${_connect}` is a probe variable that records the time required to establish a connection.

Connection duration interval: `${_active}` is a probe variable that records the duration from the connection request until the end of the end of the script.

TCP Script Commands

InterMapper supports two commands for measuring intervals during a script. These are:

STRT starts the probe's custom timer.

TIME *varname* sets the variable named `${varname}` to the milliseconds elapsed since the custom timer was started.

<script-output> Section

There is an optional **<script-output>** section in which custom TCP probes can display their results. The data in this section will appear in a popup window when you click and hold on the device. The format of this section is the same as the [<snmp-device-display>](#) section.

You may use the `${_connect}` and `${_active}` variables as well as any variables set with the **TIME *varname*** command in the **<script-output>** section of the popup window.

A Note on Accuracy

InterMapper uses different techniques to measure the round-trip times of various probes.

- **Pings (ICMP and AppleTalk echoes):** These have the most accurate timings. InterMapper is able to detect the arrival of the Ping response at interrupt time, and thus, it can compute the response times quite accurately (i.e., with millisecond accuracy).
- **Other UDP-based and TCP-based probes:** The timing on these probes are computed by the InterMapper application as it does its normal polling. Thus, the measured time can be affected by other tasks performed using the user interface, especially in the Classic MacOS. Examples of these user interactions are pulling down menus, switching applications, etc. InterMapper's timing measurements will be relatively more accurate when it's in the background, since the user interface won't interfere.

Custom SNMP Probes

The ability to construct Custom SNMP probes gives you the ability to monitor certain MIB variables that aren't tested by InterMapper's built-in probes. These MIB variables might include the CPU utilization of a router, temperature inside the equipment, switch closures, etc.

Like other probes, Custom SNMP probes are invoked and return the status and condition string for the device being tested. The basic operation of a Custom SNMP probe is:

1. InterMapper polls the device for the certain values (called *probe variables*) specified in the probe file as well as the built-in MIB variables (usually byte and packet rates for interfaces).
2. InterMapper also polls each interface for probe variables as needed.
3. InterMapper then evaluates a series of expressions in the probe file, comparing the probe variables to thresholds.
4. If a comparison is triggered (generally, if the probe variable is above or below the given threshold), then InterMapper sets the device status to the status indicated if it is "worse" than the devices current status.
5. When the user clicks and holds on a device, InterMapper processes the relevant *display* section to produce the text for the pop-up window.

Custom SNMP probes follow the same general format as other probe files. The [<header>](#), [<description>](#), and [<parameter>](#) sections work as described for other probe files.

Note: In the `<header>` section, the "type" for a Custom SNMP probe is "custom-snmp".

Custom SNMP probes have sections that describe which MIB variables to collect from the device, how those variables are to be tested against thresholds to determine the device's status, and what information about the device and its links should be displayed in the pop-up windows.

Probe Variables

The `<snmp-device-variables>` section defines values that are to be retrieved using a particular SNMP OID. These values are called *probe variables* and can then be compared to thresholds to create alarms, warnings, etc.

Each line of the `<snmp-device-variables>` section defines a particular variable to be retrieved. The definition is composed of four comma-separated attributes: the variable's name, its OID, its Type, and an optional Chart legend. Their definitions are:

Variable-name is the name used to represent the particular MIB value in this probe. A variable-name consists of letters, digits and an underscore, and must begin with a letter. Variable-names are not case-sensitive. These variable names will be represented in the probe as `${Variable-name}`. It is not necessary that a particular variable-name match the corresponding name in the MIB, although that may be convenient.

OID is the Object ID for the particular probe variable. Note that a scalar's OID must end in ".0" according to the SNMP specifications. Although it is common to off leave the suffix for scalar values, it is technically correct to include it and InterMapper requires the suffix to construct the proper queries. InterMapper will request the OID and wait for the response. [Technically, InterMapper issues an SNMP Get-Next-Request for the previous OID.] Currently, there is no facility for iterating across all rows of a particular table.

Type may be one of: Default, Per-second, Per-minute, Total-value, or Hexadecimal.

- **Default** should be used for variables of type Integer and DisplayString. InterMapper will automatically determine the type and display the value properly.
- **Per-second** and **Per-minute** will force InterMapper to compute a rate for the particular variable by computing the difference between two successive samples and dividing by the elapsed time.
- **Total-value** displays the actual value of a counter or gauge, not a computed rate value.
- **Hexadecimal** displays the data as hex digits.

Chart-legend is an optional field that provides a text label to be placed on any strip charts that incorporate this variable. Chart legends may contain variable names in the form `${variable-name}`.

Here is a sample `<snmp-device-variables>` section.

```
<snmp-device-variables>
--Variable-name      OID ---          TYPE ----   CHART LEGEND -----
ipForwDatagrams,    1.3.6.1.2.1.4.6.0, PER-SECOND, "Forwarded datagrams"
ipInHdrErrors,      1.3.6.1.2.1.4.4.0, PER-MINUTE, "IP received header err"
tcpCurrEstab,       1.3.6.1.2.1.6.9.0, DEFAULT,   "Number of TCP conn's"
sysDescr,           1.3.6.1.2.1.1.1.0, DEFAULT
</snmp-device-variables>
```

Note: The OIDs above have a trailing ".0" to specify their full OID.

Thresholds

The `<snmp-device-thresholds>` section describes the comparisons that should be made between probe variables and other values.

Each line in the threshold section is composed contains a *status*, a *comparison*, and an optional *condition string* for probe variables. If the comparison *triggers*, that is if the expression comparing the probe variable to a constant or other variable is true, then the device is changed to the corresponding status if that exceeds its current status. Here is an example:

```
<snmp-device-thresholds>
alarm: ${ipForwDatagrams} > 10 "ipForwarded datagrams too high"
alarm: ${tcpCurrEstab} >= 1
warning: ${ipForwDatagrams} > 5
warning: ${ipForwDatagrams} <= 2
warning: ${ipInHdrErrors} > 5
</snmp-device-threshold>
```

Each of the comparisons is evaluated in top to bottom order until one is triggered (e.g., is true). At that point, if the associated status is more severe than the device's current status, the device now uses its status and condition. No further comparisons are made once one has triggered.

Numeric Comparisons

In comparisons, the following numeric operators are legal: `>`, `>=`, `<`, `<=`, `=` and `!=`. The triggering comparison is also written to the log file as well as being added to the bottom of the device's popup window. If the condition string is present, it is displayed in addition to the comparison string.

String Matches

By default, InterMapper performs numeric comparisons. To compare values as strings, enclose one or both of the operands in double-quotes ("`\"`"). For example, the comparison

```
warning: ${sysContact} != "Fred Flintstone"
```

will be compared as strings because the name is enclosed in quotes. The `=~` and `!~` operators provide partial string matches. They perform "contains" and "doesn't contain" comparisons, respectively.

Popup Window Text

Custom SNMP probes can display the information gathered during polling in the popup window. Create a `<snmp-device-display>` section with the items to be displayed. Read the [Popup Windows](#) page for details.

Customizing Web Pages

InterMapper comes with a set of default web page layouts shown in Chapter 6. This appendix describes how you may customize those pages by modifying the files that InterMapper uses for creating web pages.

InterMapper's built-in web server generates pages based on files in its Web Pages directory. Whenever a web request is received, InterMapper finds a corresponding file (called a *target file*) to use as the response. The target file will be formatted according to information specified in another file called a *template file*, and then returned to the user's web browser.

The remainder of this section describes the various features of customizing the web pages. They include:

[Target files](#) which contain the main text of the various pages sent by the server

[Template files](#) which control the overall format of the web pages

[Directives](#) which are commands within files to control the formatting of the web pages

[Quoted links](#) make it easy to create links to other pages.

A [macro reference](#) section which describes each macro available

The [folder structure](#) of the Web Files directory

Tip: The target and template files are simply text files. You may edit them with any text editor.

Tip: The changes you make to these files will not take effect until InterMapper reloads these files. You may force InterMapper to reload by opening the **Preferences...** window. Choose the **Web Server** category from the popup menu, and stop and restart the Web Server to force the web pages to be reloaded.

Target Files

Whenever InterMapper receives a request for a web page, the requested URL is parsed to determine the target of the request. This *target file* contains the text that will appear as the contents of the desired page. The target file may contain HTML markup if desired. In addition to the page's text, the target file may also contain these other elements:

[Directives](#) are commands that describe or modify the way a page should be displayed.

[Quoted Links](#) provide a way to create a link to another page using its name, rather than specifying its full URL. A quoted link represented by a string in double-quotes (""): the text in quotes must match the title of another InterMapper web page.

[Macros](#) represent other text strings that will replace the macro in the final web page. The replacement text may be a static string, a device's name or network address, the contents of another file, or other information. Macros are composed of keywords and optional parameters enclosed in "\${...}".

Here is a simple target file:

```
#title "This is a test page"
```

```
This is some text to be displayed in a web page. The page's title will be "This is a test page", while the remaining text will be displayed in the "body" of the page. The text may also contain plain text, HTML tagged text such as <b>bold</b> and <i>italic</i>, and macros such as ${date} which displays today's date.
```

The first line is a directive that sets the title of the page to be displayed. The text between double-quotes will be placed in <title>...</title> tags of the resulting web page. The remainder of this example will be placed in the *body* of the resulting page. The macro `${date}` will be replaced by the current date when the page is displayed.

Note that the text "This is a test page" will be displayed as a link to its own page, since it is a string in quotes that matches the `#title` of a web page (its own). Note, too, that the text "body" could be a link to a page with a title of "body". It is not an error if no such page exists: in that case, InterMapper will simply display the quoted string in place.

As the target file is read, InterMapper processes the directives, then the expands the macros and creates the "<A HREF=..." tags for any quoted links it encounters. The web server does not insert any white space or paragraph marks (such as <P>) when it encounters carriage returns, etc.

InterMapper provides a number of built-in target files. These file names all begin with "!", and are required because InterMapper refers to them explicitly. The built-in files are:

!index.html Displays the default page, when none is specified in the URL.

!document.html Displays a graphical image of the specified map

!network.html Displays detailed information about the specified network

!device.html Displays detailed information about the specified device

!!link.html Displays detailed information about the specified link

!chart.html Displays the specified strip chart

Template Files

To allow all the web pages to have the same look, InterMapper uses *template files* to control the formatting of pages. A template file is composed of HTML commands that provide the skeleton for a web page. In addition, template files often contain macros and quoted links that will be replaced by appropriate text when the page is generated.

Here is a simple template file that could be used with InterMapper:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 3.2//EN">
<HTML>
<HEAD>
<TITLE>${pagetitle}</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
${imageref:logo.gif}
${bodytext}
${include:footer.incl}
</BODY>
</HTML>
```

This sample contains several important macros:

- `${pagetitle}` will be replaced with the text of the `#title` directive of the target file.
- `${bodytext}` will be replaced with the body text of the target file, that is, everything from the target file that is not a directive.
- `${imageref:logo.gif}` will be replaced with an `` tag that refers to the `logo.gif` file in the `~GuestImages` folder on the InterMapper server.
- `${include:footer.incl}` will be replaced with the contents of the file named `'footer.incl'`.

Directives

InterMapper has several directives that can change the way pages are formatted. Directives must start with a "#" in the first column. They are defined below:

```
#template "othertemplate.html"
```

A target file may specify a template file with the `#template` directive. The `#template` directive is optional; if none is present, InterMapper will use the file named `!template.html` as the page's template.

```
#title "This is a Test Page"
```

The text in quotes becomes the title of the page (it will be bracketed in `<title>...</title>` tags in the generated page). It also provides a destination for quoted links on other pages. Every target file must have a `#title` directive to give it a name. You may include a macro in the quoted text of this directive. This is useful for inserting the device name or other information into the title of the web page.

```
#alt_title "Test Page"
```

The optional `#alt_title` directive provides a way to give a page an alternate name that can be used with a quoted link.

```
#filename "otherpage.html"
```

The optional `#filename` directive causes InterMapper to treat the file as if it was named with the quoted string. For example, a target file named `"xindex.html"` could have a `#filename "!"index.html"` that will cause it to be used in place of the file named `!"index.html"` if its version number is higher. This can be useful for debugging as well as experimenting with alternate pages.

```
#version "2.1"
```

The optional `#version` directive determines which file will be used when there are two or more instances of the same filename (as a result of using `#filename` directives.) Version numbers must be in a 'digit.digit' format. InterMapper will use the file with the highest version number. This is useful for debugging as well as experimenting with alternate pages. The default version is "1.0" if no `#version` directive is present in the file.

```
#redirect "otherpage.html"
```

The `#redirect` directive causes the InterMapper to find `otherpage.html` and use that in place of the original target file. This can be used to force a well-known page (such as `!index.html`) to display a user-selected page.

```
#target "window_name"
```

The `#target` directive forces a page to be opened in a new window named `"window_name"`. When generating the web page, InterMapper will generate an HREF link with a `...target = "window_name" ...` reference. InterMapper uses this to force the detailed information to be displayed into a separate window after a click on a map's device or link.

Quoted Links

You can create links to other pages simply by entering the page's title in double-quotes. For example, the text "Test Page" will create a link to the page with a `#title` or `#alt_title` directive that contains "Test Page".

Note that two target files may have the same `#title` or `#alt_title`. If so, InterMapper will choose one of those target files, but the one that will be chosen is not predictable.

To prevent a string in quotes from being interpreted as a quoted link, you should place backslashes ("\") in front of *both* the first and second quotes.

It is not an error to have a quoted string that does not match another page's `#title` or `#alt_title`. In that case, InterMapper will simply display the quoted string as-is.

Macro Reference

A macro is a text string with the format `${macroname:other-information}`. The *macroname* is required, and some macros take *other-information* which follows the ":". The entire macro will be replaced by the appropriate text when the page is generated.

Macros that generate the "content" of an InterMapper web page

InterMapper often uses these macros either as the `${bodytext}` of the page, or as a major part of a page's contents. All the macros below work on the map named in the request URL. If the URL is for a page in the `~/admin` directory, InterMapper will display information about all the items in all maps.

`${fullstatus}` shows a list of of all the devices and links for the map named in the URL.

`${errorstatus}` shows only the devices and links which are in warning or alarm states, or down for the named map.

`${currentoutages}` shows the list of current outages -- devices or links that are currently in warning, alarm, or are down in the named map.

`${previousoutages}` shows the list of devices that had been listed as outages but have since returned to normal.

`${maplist}` shows an HTML unnumbered list () of the maps available.

`${chartlist}` shows an HTML unnumbered list of charts for the map.

`${maplistwithcharts}` shows an HTML unnumbered list of the maps available, with sub-lists of the charts for each map

Miscellaneous macros that describe InterMapper and its environment

`${abouthtml}` shows the "About" page with the current version of InterMapper.

`${statshtml}` shows InterMapper's statistics: its uptime, memory usage, etc.

`${httpuserid}` the name the user typed when asked for authentication

`${httpremoteaddress}` the IP address of the remote browser.

`${intermapperaddress}` the IP address of this InterMapper server

`${version}` the version of this copy of InterMapper

`${date}` the current date

`${time}` the current time

`${imagesuffix}` set to ".png" if the web client can display PNG images, or ".jpeg" otherwise.

`${telnetserverurl}` a telnet: URL that will connect to this InterMapper server

`${webserverurl}` a http: URL that will connect to this InterMapper server

`${mapname}` the current map's name

`${deviceaddress}` the IP or AppleTalk address of the particular device. It is an empty string for anything that isn't a device.

`${devicename}` the DNS name or AppleTalk NBP name of the particular device. It is an empty string for anything that isn't a device.

`${pagetitle}` displays the value set by the `#title` directive

`${SetNameFieldWidth:xx}` Set the width of the name field. InterMapper pads the name up to xx characters wide. Use -1 for "only as long as it takes". The default width is 20 chars.

Include Macro.Your template files and target files may include other files.

`${include:file-to-be-included.html}` the named file will be inserted into the web page. The file must be in the same folder.

Macros to place images onto a page.

`${imageref:imagefile}` creates an `` tag to place an image on the page

`${imagesuffix}` returns ".png" or ".jpeg", depending on the file format the web browser supports.

`${intermapperlogo}` creates an `<img... >` tag that includes the "Made with InterMapper" logo.

Macros that control the interval between page refreshes.

InterMapper's web server can automatically refresh a particular web page at a desired interval. Include these tags on your page to take advantage of this facility.

`{htmlrefreshmetatag}` is either an empty string or the previous refresh choice from the web client. (Inserts a `<meta http-equiv="refresh"...">` tag on the resulting page.)

`{htmlrefreshmetaoptions}` is the option list that a web client can choose from. The current `{htmlrefreshmetatag}` value is selected. Note that your HTML template should supply the `<form><select>...</select></form>` surrounding this `{htmlrefreshmetaoptions}` macro.

Macros that describe the requested URL

These macros all return a fully-escaped string, that is, a space character (" ") will be replaced with a %20; a "?" with %3F; etc.

Here is a sample URL. The result of using this URL is shown in parentheses after each macro:

`http://localhost/Map1/device/192.168.0.1%3ASNMP/!device.html`

`{webpageurl}` the full URL of the requested web page. (the full URL as shown above)

`{httppath}` the full path to the file requested
("/Map1/device/192.168.0.1%3ASNMP/!device.html")

`{httpdocument}` the top level directory of the page requested. Also an alias for `{mapname}`
("Map1")

`{httpclass}` the second level directory of the page requested (device, chart, link, document, network) ("device")

`{httpinstance}` the third level directory of the page requested("192.168.0.1%3ASNMP")

`{httpmethod}` the fourth-level part of the page requested ("!device.html")

`{httpinstancepath}` a concatenation of `{httpdocument}`, `{httpclass}`, `{instance}` separated by "/"
("/Map1/device/192.168.0.1%3ASNMP")

Folder Structure

The web target files and template files are in the *Web Pages* folder within the *InterMapper Settings* folder. The InterMapper web server will serve out files at the top level of the *Web Pages* folder. It will not serve out files from folders at the top level except those described below.

InterMapper ships with four folders stored there:

~AdminHTML

This folder contains HTML templates for pages that show the overall status of the InterMapper program. People who have access to these pages may also view all the separate map pages. These files may be accessed from the default web URL, or by using a URL in this form:

`http://intermapper.domainname.com/~admin/filename.html`

~GuestHTML

This folder contains HTML templates for reporting errors such as missing or invalid file names, or responding to web clients who are not authorized for the web server. These files bypass the usual access list mechanism and may be accessed using this URL form:

`http://intermapper.domainname.com/~error/filename.html`

~GuestImages

This folder contains images used by the InterMapper web server. These images may be placed in a target or template file using the `#{imageref: ... }` macro.

PerMapHTML

This folder contains HTML templates for displaying a map's information. To customize the HTML for a particular map, duplicate the PerMapHTML folder, and edit the file(s) within that folder. You must also select the new folder name from the popup menu in the map's [Map Settings...](#) dialog. To view a specific document's information, use this URL form:

`http://intermapper.domainname.com/docname`

The main web page is the `~admin/!index.html` target file. When an unqualified URL request arrives (that is, a request for `/`, without any additional path of file information), InterMapper sends out the file specified by `~admin/!index.html`.

By default, all target files use the same template, `!template.html` (note the exclamation point at the beginning of the filename). A target file may specify a different template file by using the `#template` directive.

For both the `"~AdminHTML"` and `"PerMapHTML"` folders, the default HTML page is `!index.html`. A request for `http://intermapper.domain.com/` will be treated like a request for `http://intermapper.domain.com/~admin/!index.html`.

Similarly, a request for `http://intermapper.domain.com/docname` will be treated like a request for `http://intermapper.domain.com/docname/document/main/!index.html`.

InterMapper Files and Folders

InterMapper saves its files in specific folders. In particular, the following file and folders have special locations.

- Map files
- Settings folder
- Log files

Note: The locations of these files and folders differ slightly between MacOS Classic and X. See the [Classic/MacOS X differences](#) later on this page.

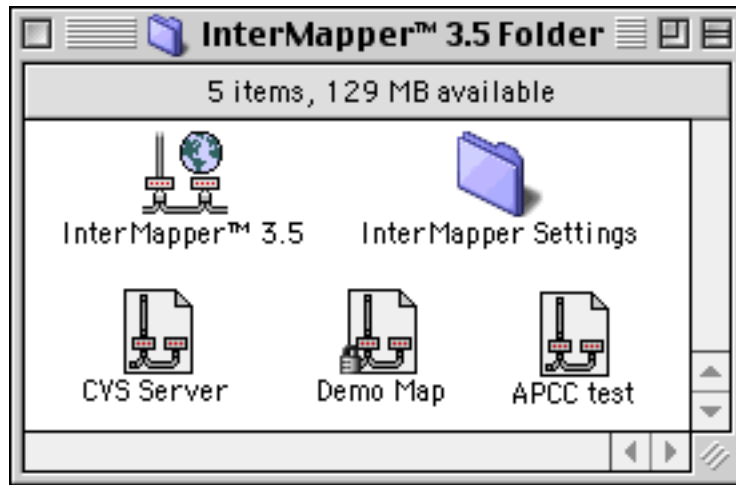


Figure D-1: A typical InterMapper top-level folder. It shows the InterMapper application, three map files and the settings folder.

Map Files

InterMapper will automatically open all the map files that are contained in its folder when it starts up. Thus, it is a good idea to save all the maps you create in the top-level InterMapper folder. Figure D-1 shows the typical top-level folder for InterMapper.

Settings Folder

All InterMapper settings are kept in an *InterMapper Settings* folder as shown. The contents of this folder are described in detail in the [Settings Folder](#) page.

Differences between Classic MacOS and MacOS X

The placement of the folders vary slightly between the Classic MacOS and MacOS X versions. The *InterMapper Folder* described below contains the InterMapper application. Any map files enclosed in that folder will be opened when InterMapper launches.

	Classic Mac OS	Mac OS X
<i>InterMapper Folder</i>	Anywhere on the hard drive	Within the user's <i>Applications</i> folder
<i>Map Files</i>	Within the top-level <i>InterMapper Folder</i>	Within the top-level <i>InterMapper Folder</i>
<i>InterMapper Settings</i>	Within the top-level <i>InterMapper Folder</i>	In <i>/Library/Application Support</i> folder

InterMapper Settings

The *InterMapper Settings* folder contains all the settings, preferences, and configuration of InterMapper.

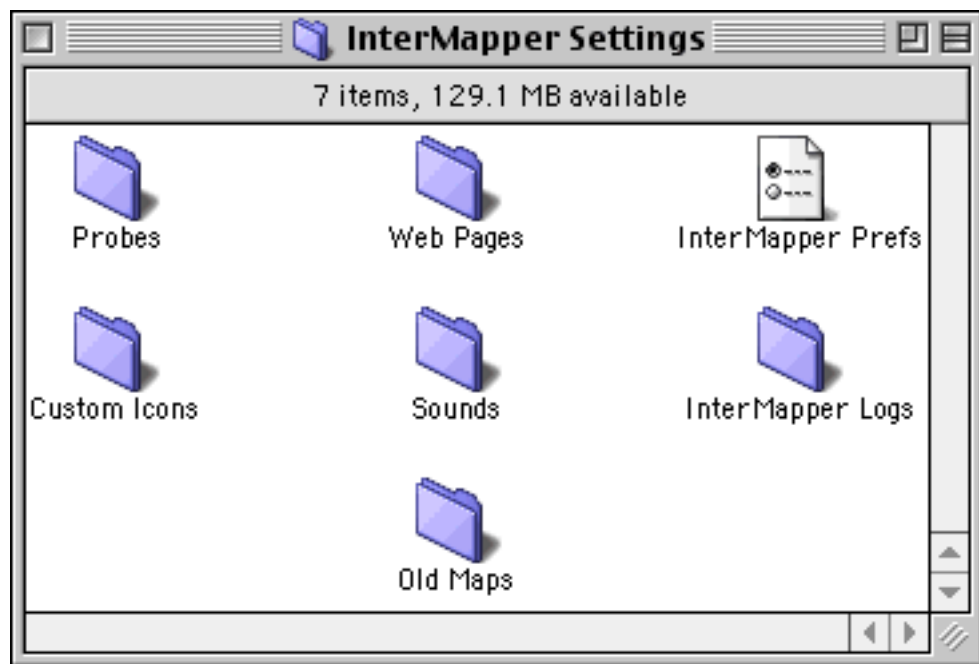


Figure D-2: The *InterMapper Settings* folder.

This folder contains the following items:

- **Probes:** This folder contains built-in and custom probes. Probes are text files that add functionality to InterMapper so that it can test new devices. See [Appendix B -- Customizing InterMapper's Probes](#) for details about creating and customizing probes.
- **Web Pages:** This folder contains the template and target files that describe the web pages that the InterMapper server displays. See [Appendix C -- Customizing Web Pages](#) for details about customizing these pages.
- **InterMapper Prefs:** This file contains the current settings of all the preferences.
- **Custom Icons:** You may add custom icons to maps to enhance InterMapper's built-in set. See [Appendix D -- Custom Icons](#) for details about making and adding custom icons.
- **Sounds:** Add .aiff, .mp3, etc. files to this folder to make them available for InterMapper notifications.
- **Chart Data:** This folder contains the saved data of strip charts. When InterMapper launches, it reads the data from these files to restore the strip charts history.
- **InterMapper Logs:** This folder contains text files that log events that InterMapper has detected. Previous versions of InterMapper kept their log files in the *InterMapper* folder itself. InterMapper 3.5 places new log files into this folder within the *InterMapper Settings* folder.
- **Old Maps:** This folder contains unmodified versions of maps that were created by previous versions of InterMapper. Whenever a new version of InterMapper requires a map file format change, InterMapper will save an exact copy of the earlier map in this *Old Maps* folder before saving the operational map in the new format. This makes it simple to revert to an earlier version of InterMapper - simply restore the old map file from the *Old Maps* folder and launch the previous version.
- **InterMapper User List:** Previous versions of InterMapper kept the user list in a separate file. With InterMapper 3.5, these user and group settings have been incorporated into the *InterMapper Prefs* file. You may leave this file in place without affecting InterMapper's operation.

Custom Icons

InterMapper provides a facility for adding custom icons into your maps. The files containing the icons must be placed in the *Custom Icons* folder inside the *InterMapper Settings* folder.

There are certain restrictions to the use of custom icons:

- The icons must be of type 'ICN#'. InterMapper supports 32 x 32 black and white icons only.
- The icons must have resource ID's greater than 0.
- As a rule, custom icon ID's should be greater than 5000, so they won't conflict with InterMapper built-in icons.

A resource file containing icons can have any creator code or file type. Resources other than 'ICN#' contained in the icon file are ignored.

The Custom Icon Gathering Project

Jakob Peterhønsel has assembled a number of icons for use with InterMapper. They may be found at <http://www.hjemme.dk/tcigp/>. If you design any icons, please consider contributing them to Jakob's collection.

Making Backups

InterMapper saves its state in two folders: the main *InterMapper folder*, and the *InterMapper Settings* folder.

As described in the [Files and Folders](#) page, InterMapper saves these in different places for Mac Classic and Mac OS X.

- To make a backup of InterMapper on **MacOS Classic**, simply back up the main InterMapper folder. The InterMapper Settings folder is contained within it, and will be backed up automatically.
- To backup InterMapper on **MacOS X**, you should back up the main InterMapper folder and the InterMapper Settings folder in the `/Library/Application Support/` folder.

Frequently Asked Questions

This Appendix lists frequently asked questions about InterMapper. It is divided into the following sections:

[Overview](#)

An overview of questions asked about InterMapper.

[Setting Up InterMapper](#)

A quick introduction to installing and getting started with InterMapper. You should also read the [Tutorial](#) section of this manual for details.

[Understanding the Map](#)

These questions give a deeper understanding of everything that's represented on the map.

[Troubleshooting Tips](#)

What can you do to make the map look better?

[What is SNMP?](#)

A brief description of SNMP (the Simple Network Management Protocol) and links to more information.

[Probe Types](#)

Here are some technical details about how InterMapper probes and tests devices.

[Miscellaneous](#)

A grab-bag of questions that don't fit elsewhere...

[Problems with Open Transport SNMP](#)

Details on the problem with Open Transport SNMP for the Mac

[Dealing with unmanaged hubs](#)

Many customers ask how they can connect devices to the proper hub, even if it's an un-managed (e.g., "dumb") hub. This page shows how.

[Changing Port Labels](#)

A switch's ports have labels. Read this to find out how to change them to be more informative.

[Newly Answered Questions](#)

These questions haven't been filed into other pages yet. Come back again to see what we've added...

FAQ: Overview

- [What is InterMapper?](#)
- [What do I need to run InterMapper?](#)
- [I'd like to try it out. How can I get a demo version?](#)

What is InterMapper?

InterMapper is a network and server monitoring and alerting tool. Its primary focus is IP and AppleTalk based equipment. It provides a graphical map view of the network, and can autodiscover the topology of the network if SNMP-speaking routers are present. It can also scan the subnets on the map to find devices connected to that network.

After a map has been created, InterMapper monitors the network. It queries routers, switches, and hubs with SNMP to get traffic statistics and displays the data on the map as little "ants" marching along the links. (Our customers love that!) The network manager can also use the bundled SNMP Watcher program to get even more detailed information about a device.

InterMapper also polls various kinds of servers, including Web, FTP, Mail services (SMTP, POP3, IMAP), LDAP, RADIUS, DNS, RTSP, QuickTime Streaming, and AppleTalk Echo, RTMP, AppleShareIP, and KeyServer. InterMapper will detect a lack of response, or an incorrect response, and send a notification.

When InterMapper detects a problem, it marks the device on the map in red, orange, or yellow to indicate the severity. It can also send notifications as audible alerts, e-mail messages, pages, or it will run programs on the computer to initiate corrective action. These notifications may be sent according to a schedule, and notifications can be escalated if a situation continues unacknowledged for a longer time.

InterMapper also has a built-in web server that allows network managers to view the status of the network from an ordinary web browser. There is also an access control mechanism which allows the network manager to give privileges to view certain maps to designated individuals. (This was requested by our ISP customers whose own clients wanted to see their maps.) The web server can display both the graphical maps shown on InterMapper's screen as well as concise text-based summaries of the network's operation.

And last, InterMapper is delightfully easy to set up. (We know -- every vendor says this. But it's really true!) Because of its autodiscovery features, it's easy to build a map. And our five years of development in this program have given us a chance to refine the user interface to remove the rough edges while providing the essential features that people require.

[Back to top](#)

Do I need a powerful computer to run InterMapper?

Not really. Nearly any Macintosh computer manufactured in the last five years will do nicely for all but the largest networks. In fact, our production machine is a Quadra 700 with 20 MBytes of RAM. We recommend you have:

- A Macintosh computer with at least 16 MB of RAM.
- System 7.5 or greater.
- Open Transport 1.1.2 or later
- A properly configured network connection (LocalTalk, Ethernet, ARA, SLIP, PPP, etc.).

You'll find InterMapper most useful if you have a network with a few SNMP-speaking routers. InterMapper is less visual for so-called "flat" networks without routers, but it will prove useful for monitoring hosts by sending them ICMP or AppleTalk echo packets.

[Back to top](#)

Is a demonstration version of InterMapper available?

Yes. There's a link on all of the pages of the InterMapper web site (<http://www.intermapper.com>) to download the demo. You'll need to get an evaluation serial number to use the program for longer than an hour. There's a form on the demo page that you can use to request one.

FAQ: Setting up InterMapper

- [How do I map my network?](#)
- [How far does auto-discovery go?](#)
- [My router does not answer SNMP...](#)
- [How do I edit the map?](#)
- [How do I smooch things out flat? \(ie. automatic layout\)](#)
- [There are two separate network bubbles on my map where there should only be one...](#)
- [Some network bubbles have more than one IP network number...](#)
- [Is there any way to hide some of the detail?](#)
- [How do I bring hidden stuff back?](#)
- [Does InterMapper support unnumbered IP links?](#)
- [Is there another way to add many devices to a map?](#)

How do I map my network?

Whenever you create a new InterMapper map, you will see the following dialog. This allows you to specify the starting point for auto-discovery.

By default, InterMapper fills this field with the address of your own computer. There is also a checkbox which specifies how many hops from the starting point InterMapper should auto-discover, and a field for entering an SNMP community string (you can try "public" unless you've configured your routers otherwise.) Press Return and InterMapper will begin discovery.

For more information about setting up the map, read the [Tutorial section](#) of this manual.



[Back to top](#)

My router does not answer SNMP...

InterMapper uses SNMP to learn about attached networks and create the network clouds you see on the map. When a device doesn't answer SNMP, InterMapper will automatically probe it with the standard echo probes, so you can still monitor the device.

Here are some reasons your routers might not answer:

- The device might not respond to SNMP queries with a community string of "public".
- There might not be an SNMP agent running in the device.
- The device might not be reachable.
- The Mac running InterMapper might not be entered in the device's access list.

A **community string** is akin to a login-name for a router. SNMP uses community strings to specify what variables a client can read or modify. A network manager often sets a community string of "public" to permit read-only access to SNMP data. If your site uses a different community string for read-only access, you must specify this when adding devices or the device will not respond.

[Back to top](#)

How do I edit the map?

This is a quick overview of editing the map. Also read the [Arranging the Map](#) section of the tutorial.

- Before you can make changes to the map, press *Tab*, select "Map Editor" from the "Edit" menu, or click on the pencil icon in the lower left-hand corner. When the slash disappears, the map is editable.
- To move a router or network, select and drag it.
- To change the shape or label characteristics of an item, select the node and choose the appropriate command from the "Display" menu.
- To resize a wire-shaped network, click and drag the endpoints.
- To select a node and its adjacent nodes, option-click on a vertex. Continuing to option-click continues to select adjacent items.
- To select the devices at each end of a link, option-click on the link.

Summary of selection tricks:

- Shift-click to select multiple items.
- Option-click to select adjacent nodes or endpoints of a link.
- Option-click after layout to select the subtree rooted at a node.
- Use the "Select Other" sub-menu under "Edit" to select all routers, networks or "leaf nodes".

[Back to top](#)

How do I smoosh things out flat? (i.e. automatic layout)

Try one of the layout commands from the "Layout" menu. If you don't see a Layout menu, click on the pencil icon in the lower-left corner.

- Cycle: This moves all selected devices to the edge of the window. This is most useful when doing initial layout. Once the devices have been positioned by the Cycle command, you can see the most heavily interconnected parts of the network.
- Bus: This moves all devices which are connected to the selected network into a "bus" layout. InterMapper draws a vertical line, and then positions each attached device close to that network.
- Star: This moves all the devices into a circle near the the selected network.

You can then use commands from the Display and Layout menus to change items position or appearance on the map.

[Back to top](#)

There are two separate network ovals on my map where I only expect one...

Examine the network popup (click and hold on a network) to determine whether the subnet masks are the same in both bubbles. If the subnet masks are different, one of the devices connected to the bubble with the "wrong" subnet mask probably has a misconfigured subnet mask. (Look for the device that is being polled with SNMP.)

Note: For devices polled with ICMP echoes, InterMapper tries to guess whether it should draw a link to the network that contains the IP address. If both network bubbles look equally good, it may draw a link to the "wrong" one, or alternate between them.

[Back to top](#)

Some network ovals have more than one IP network number...

It's possible for a router or host to have two or more configured IP addresses for a particular interface. This form of secondary IP addressing can be common if your addressing is in transition. Rather than bringing everything to a halt to change IP addresses, a network administrator will support two IP subnets on the same

logical wire. All the devices in the subnet can then have their IP addresses changed at their leisure, rather than forcing everyone to change them all at once. When all the addresses have changed, the administrator will usually get rid of the old network number.

It's also possible that InterMapper is only reporting what it knows, and the information it is using is incomplete. This may be true of multi-point network technologies (like frame-relay clouds). If you find a situation where InterMapper is reporting multiple networks on a logical network and you know it's wrong, please send us mail (InterMapper@dartware.com) so we can figure out a way to make InterMapper's depictions more accurate.

We would also like to hear about a network with multiple IP network numbers where InterMapper does not show them correctly.

[Back to top](#)

Is there any way to hide some of the detail?

Select the networks or devices you don't want to see and choose "Hide" (Edit menu).

InterMapper will not poll hidden interfaces.

[Back to top](#)

How do I bring hidden stuff back?

You have two options. First, you can use the "Show All" command. This will make visible everything you have hidden. If you have hidden a lot of routers and links, you can select a device and choose "Show Adjacent" to make only the adjacent elements visible.

[Back to top](#)

Does InterMapper support unnumbered IP links?

Yes.

[Back to top](#)

Is there another way to add many devices to a map?

There are several ways to do this:

1. Use the **Add Devices...** command as described in the [Adding Devices](#) section of the Tutorial.
2. Copy device names from MacPing 3.0 and paste them into InterMapper. (You can find a 30-day demo version at the <http://www.dartware.com/macping/> website.)

MacPing stores the information on the clipboard as text. You can also create configuration files using your favorite text editor and then copy and paste from that.

The format for AppleTalk addresses is a tab-delimited line of the form:

```
<object><TAB><type><TAB><zone><TAB><ddp-address><RETURN>
```

The format for IP addresses is a line of the form:

```
<domain-name><TAB><anything><TAB><ip-address><RETURN>
```

You can copy as many lines as you want. Empty lines are ignored.

3. You can Copy a number of DNS names or IP addresses, one per line, and paste them into a map. They will be added into the map, as described in the [Importing Devices](#) page.

[Back to top](#)

FAQ: Understanding the Map

- [What do the "moving ants" represent?](#)
- [What do the boxes and bubbles represent?](#)
- [What do the lines represent?](#)
- [What do the colors mean?](#)
- [What does an X on a link mean?](#)
- [What does the "Device List" show?](#)

What do the "moving ants" represent?

InterMapper draws dotted lines ("ants") next to a link to indicate that its current traffic flow is above a user-settable value. Use the **Traffic Thresholds...** command under the **Network** menu to change the settings and for a legend of the different varieties of ants. You only see the ants in "browse" mode (as opposed to "edit" mode.) To toggle between the two modes, press the *Tab* key or click on the pencil icon in the lower left-hand corner.

InterMapper regularly polls all the visible interfaces for packets, bytes, errors and discards.

[Back to top](#)

What do the boxes and ovals represent?

The *boxes* represent the physical equipment of your network. The *ovals* represent the networks which link the routers together. The numbers in the bubbles are "network identifiers". For IP networks, the number is the network and the subnet portion of the IP addresses of all devices on it. For example, "192.0.16.0/24" is a network where IP addresses are in the range 192.0.16.0-192.0.16.254, and the subnet mask has 24 bits e.g., it is a Class C network). This is described in detail in the [Subnet Mask](#) FAQ.








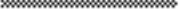

For AppleTalk networks, the number is the AppleTalk network number.

If you click and hold on a router or network you will see a popup window with information about that item. (This only works in "browse" mode -- click on the pencil icon in the lower left corner till it has the slash through it).

[Back to top](#)

What do the line styles represent?

The style of the line corresponds to the type of interface.

	10 Mbit Ethernet
	100 Mbit Ethernet or FDDI
	Serial line - T3 Speed
	Serial line - T1 Speed
	Serial line - 56 K or other
	Frame-Relay Interface Type
	ATM Interface Type
	LocalTalk Interface Type
	Any type not specifically represented

As with the networks and devices, you can click and hold on an link to see a popup window with information about the interface type and traffic statistics.

[Back to top](#)

What do the colors mean?

Devices turn different colors depending on the magnitude of the problem detected. Links may be "haloed" with yellow or orange as utilization reaches 50 and 90 percent respectively.

Green

The device is up and nothing is wrong.

Yellow

The device is in warning. There may be errors on one or more interfaces.

Yellow haloes over links indicate that utilization is greater than 50% of the total capacity.

Orange

The device is in alarm. There may be more errors on an interface.

Orange haloes over links indicate that utilization is greater than 90% of the total capacity.

Red (Flashing)

The device is not reachable and is probably down. (A device may be blinking red if it is "behind" another router or link that has failed without an alternate path.)

Gray

The device's status is unknown or not being polled.

Purple

The device is searching for adjacent routers (discovery) or tracking down unnumbered interfaces.

These are the default color assignments. You can redefine the colors from the [Preferences](#) window.

[Back to top](#)

What does an X on a link mean?

An X in the middle of a link means the link is down. A red X signifies that the link's "operational status" is down (e.g., it is probably broken). A blue X signifies that the link's "administrative status" is down (e.g., it has been explicitly disabled.)

[Back to top](#)

What does the "Device List" show?

The device list summarizes the status of all devices. It contains a list of the devices in all the open maps. You

can sort any column by clicking a label at the top. The order of devices will change when their network status changes.

Double clicking on a device will bring the associated map to the front.

You can resize the columns by option-clicking and dragging any of the labels at the top. Column size is automatically stored as a preference.

[Back to top](#)

FAQ: About IP Addresses

- [What is an IP address? How do I get one?](#)
- [How do computers send data through the Internet?](#)
- [What is a subnet? Why do I care?](#)
- [What does the "/24" mean? How does that relate to my subnet mask?](#)

What is an IP address? How do I get one?

An IP address ("Internet Protocol address") is a number that represents a single unique computer on the Internet. IP addresses are similar to telephone numbers, in that each computer (or telephone) must have its own unique IP address (telephone number.) Like telephones, there's a directory system - called the Domain Name System, or "DNS" - that can convert a name such as "www.apple.com" into a corresponding numeric IP address.

IP Addresses are written as a sequence of four numbers separated by ".", like this: 208.123.246.35. Each of the four numbers in the IP address can take the value between 0 and 255.

Every computer on the Internet must have a unique IP address. ISPs purchase large blocks of consecutive IP addresses, and then allocate smaller ranges of these addresses to their customers. Thus, a particular company might be assigned all the 254 IP addresses in the range 208.123.246.1 to 208.123.246.254. (The addresses ".0" and ".255" are not usually assigned.) Companies then assign the IP address to individual computers within the organization.

[Back to top](#)

How do computers send data through the Internet?

Computers send information through the Internet by dividing the data to send into small chunks ("packets") and transmitting them to the other device. All this happens without your doing anything - the web browser, e-mail program, etc. all take care of these low level details.

When your computer wants to send to another computer, it creates the packet, then places the other computer's address in the destination address of the packet, places its own address in the source address of the packet, and then sends the packet off, either directly to the destination computer, or to a nearby router that takes responsibility for routing the packet.

There's an analogy here with the post office here. Packets are like envelopes, with destination addresses and return addresses. Routers are like post offices: they check the destination address and have the responsibility for delivering the packet to the final destination computer or to another router that's closer to the destination.

[Back to top](#)

What is a subnet? Why do I care?

A *subnet* is a range of IP addresses. The special feature is that all the computers within a subnet (a "sub-network") can all talk directly to each other, and don't need a router to communicate.

As mentioned above, your computer delivers a packet directly to the destination computer or sends it to the router for ultimate delivery.

But how does your computer know whether the packet's destination is within its subnet? The answer is that your computer uses the subnet mask to determine the members of the subnet.

There's a chart below that associates the number of IP addresses in a subnet to the subnet mask. For example, the subnet mask "255.255.255.0" represents 254 consecutive IP addresses. If your computer's IP and the destination computer's IP addresses are in the same subnet address range, then they can send

packets directly to each other. If they're not in the same range, then they must send their data through a router for delivery.

[Back to top](#)

What does the "/24" mean? How does that relate to my subnet mask?

InterMapper uses a shorthand notation to represent an IP subnet's information. The number in the "/xx" shorthand stands for the number of bits (technically, bits set to one) in the subnet mask. The convention is always to start at the left end of the 32-bit subnet mask. The table below shows the correspondence between the "/xx" notation and the actual numeric representation.

Subnet Mask	# of Addresses	Subnet Mask	# of Addresses
/1 128.0.0.0	2.1 billion (<i>Class A</i>)	/17 255.255.128.0	32,766
/2 192.0.0.0	1 billion	/18 255.255.192.0	16,382
/3 224.0.0.0	536 million	/19 255.255.224.0	8,190
/4 240.0.0.0	268 million	/20 255.255.240.0	4,094
/5 248.0.0.0	134 million	/21 255.255.248.0	2,046
/6 252.0.0.0	67 million	/22 255.255.252.0	1,022
/7 254.0.0.0	34 million	/23 255.255.254.0	510
/8 255.0.0.0	17 million	/24 255.255.255.0	254 (<i>Class C</i>)
/9 255.128.0.0	8.4 million	/25 255.255.255.128	126
/10 255.192.0.0	4.2 million	/26 255.255.255.192	62
/11 255.224.0.0	2.1 million	/27 255.255.255.224	30
/12 255.240.0.0	1 million	/28 255.255.255.240	14
/13 255.248.0.0	524 thousand	/29 255.255.255.248	6
/14 255.252.0.0	262 thousand	/30 255.255.255.252	2
/15 255.254.0.0	131 thousand	/31 255.255.255.254	RFC 3021
/16 255.255.0.0	65,534 (<i>Class B</i>)	/32 255.255.255.255.	<i>Loopback address</i>

[Back to top](#)

FAQ: Troubleshooting the Map

- [How do I change the community string?](#)
- [Why doesn't InterMapper find AppleTalk networks?](#)
- [I discovered a multi-protocol router \(TCP/IP & AppleTalk\) using TCP/IP, but I could not go back and change the protocol to RTMP?](#)
- [How do I view AppleTalk zone names?](#)
- [How do I monitor a fixed IP \(or AppleTalk\) address?](#)
- [What is "dynamic AppleTalk and IP addressing"?](#)
- [I still can't make my router talk...](#)
- [My switches are always orange and showing lots of errors \(or discards\). Why?](#)
- [What does it mean when InterMapper says a "subnet mask is discontinuous"?](#)
- [Why do network labels sometimes have a "/2*"?](#)

How do I change the community string?

You can open the **Get Info** window on a device as described in the [Device Info window](#) page.

To set the community without opening a Get Info window, select the device and choose **Set Info** from the "Network" menu. This hierarchical menu will pop up; select Community. This also allows you to set the Read-Only community string for many devices at once.

To view the community string for a specific device, select the device and use the "Get Info..." command. Click on the "More" button in the lower right corner to expand the window.

[Back to top](#)

Why doesn't InterMapper find AppleTalk networks?

To learn about AppleTalk networks, InterMapper asks SNMP-managed devices for the list of ports from their AppleTalk MIB. If the router does not support the AppleTalk MIB, InterMapper will not display the AppleTalk network numbers even if the device routes AppleTalk. Check the version number of the routing software and the router documentation to see if it supports the AppleTalk MIB.

[Back to top](#)

I discovered a multi-protocol router (TCP/IP & AppleTalk) using TCP/IP, but I could not go back and change the protocol to RTMP?

This problem is related to having only one "target" address for each device, even though InterMapper knows the addresses of the other ports. When a device is discovered using TCP/IP, it gets added with a target address in IP. You can't switch to use RTMP because that's an AppleTalk-only protocol.

[Back to top](#)

How do I view AppleTalk zone names?

InterMapper does not support this.

[Back to top](#)

How do I monitor a fixed IP (or AppleTalk) address?

In the "Add Device..." dialog, enter an IP address in dotted-decimal notation or an AppleTalk address in the form "net/node" (decimal notation).

IP addresses discovered using the IP discovery feature are fixed by default.

[Back to top](#)

What is "dynamic AppleTalk and IP addressing"?

AppleTalk addresses are inherently dynamic. Every time an AppleTalk device starts up, it may choose a different address on the network. IP addresses tend to be more stable, but with the advent of server-based IP addressing schemes (such as DHCP, BootP, MacIP, etc.), it's also possible that your IP address could change every time you start up.

AppleTalk solves the problem of providing a stable service name for AppleTalk devices by using the Name Binding Protocol (NBP). A client wishing to determine the current address of some service sends a special probe to the zone where the device is known to reside. Likewise, InterMapper periodically makes sure that it is probing the right address and immediately rechecks if the device goes down.

On the Internet, stable name identifiers are provided by the Domain Name System. InterMapper will periodically resolve a device's domain name to check for changes. If it can't resolve the name to an address, it will report the device down. The period for checking DNS names is derived from the TTL in replying DNR messages.

I still can't make my router talk...

If you still can't make the router work with InterMapper, try the following:

- Type cmd-option-shift-Z to bring up InterMapper's debug window. Look for any messages related to that device.
- Use MacPing 3.0 to determine whether the device will respond to simple, single variable SNMP requests. Use the "Test IP..." function and set the probe type to "SNMP Request". You can find the MacPing demo on www.dartware.com.
- Let us know. Send E-Mail to InterMapper@dartware.com with information about the type of device and the trouble you're having.

[Back to top](#)

My switches are always orange and showing lots of errors (or discards). Why?

We frequently hear of devices that appear to have high levels of discards and/or errors. They are usually orange on the map, and the popup window shows a message like this:

```
Reason: Discards = 738: [1] sc0
```

The most likely reason that InterMapper shows a high rate of discards from a device is that the device is actually reporting these errors. It's common that when InterMapper reports errors (from its SNMP queries), the manufacturers' own monitoring tools will report zero errors. (It's also normal that the affected devices are operating normally, without problems, in this state.)

Experiments and Workarounds:

1. You should use the vendor's own network monitoring tool (e.g., by telnetting in, using a web browser, etc.) to see if errors are being reported through their native management interface. It's possible that there *might* actually be a problem.
2. This may be a bug in the SNMP implementation of the device. You can check with your vendor to see if there's a firmware upgrade that may address this.
3. To test InterMapper's accuracy, you can use another SNMP console (for example, with SNMP Watcher, <http://www.snmpwatcher.com/>) to check out the particular MIB variables for the device. InterMapper

monitors the ifInDiscards and ifInErrors MIB variables (and the corresponding ifOutxxxx variables) listed on the [Network and Server Probes](#) page to compute its error & discard figures.

You can monitor these same variables with SNMP Watcher or your own SNMP Console to see if the same errors are reported there.

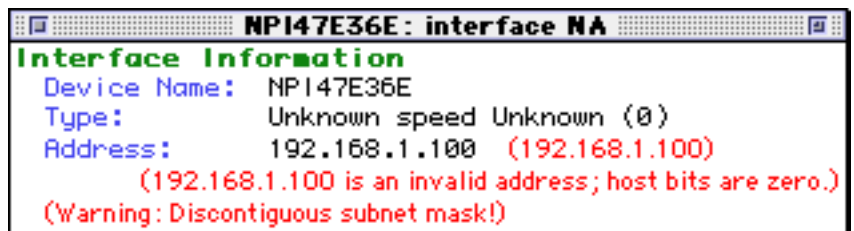
4. You can run a ping test through the device that's reporting the errors. If packets are actually being discarded, you'll see a higher than normal packet rate of dropped packets. (On the other hand, if packets aren't being dropped, then it's another clue that the values reported by SNMP are incorrect.)
5. As a workaround, if you've satisfied yourself that the error reports are bogus, you can instruct InterMapper to ignore the discards and/or errors. To do this, Get Info on the affected device and check the "Ignore Interface Errors" or "Ignore Interface Discards" box as desired.

[Back to top](#)

What does it mean when InterMapper says a "subnet mask is discontinuous"?

In usual network configurations, a device's subnet mask contains one bits in the left side of the number, and zero bits on the right. InterMapper can then use the convention that a subnet mask is described as the number of bits in the subnet mask, and uses the notation of "/24" to indicate a subnet mask of 24 one-bits, or "255.255.255.0". For more details, see the [IP Addressing FAQ](#).

A subnet mask that has zero bits interspersed with the one bits in the left half of the value is often a configuration error. InterMapper points this out when you click and hold on a link: the popup window will resemble the figure at the right.

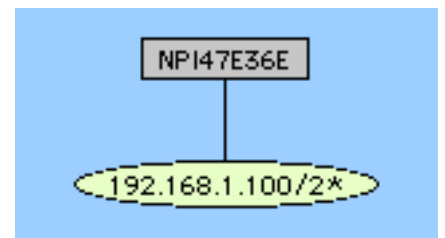


Normally, the address line contains the IP address and the subnet mask. This example shows a device whose IP address and subnet mask are set to the same value. This error is shown in the popup window.

[Back to top](#)

Why do network labels sometimes have a "/2*"?

This is another indication that there's a problem with the subnet mask. The figure at the right shows the network oval with a discontinuous subnet mask. The /2* indicates that the subnet mask has zero bits in the left half; clicking on the link will give a popup window similar to the one above.



This figure comes from an HP printer that has a bug in its SNMP implementation. The subnet mask of the printer is actually configured right, and the printer is working properly. However, the SNMP software in the printer is reporting the incorrect value (it's reporting the IP address) for the subnet mask. Dartware has reported this to HP.

[Back to top](#)

FAQ: SNMP Information

- [What is SNMP?](#)
- [What is the 'Read-only Community String'?](#)
- [Why can't I get SNMP information from a device?](#)
- [How can InterMapper query a particular MIB variable?](#)
- [Do all tables have an index?](#)
- [Where can I read more information about SNMP?](#)

What is SNMP?

SNMP stands for the Simple Network Management Protocol. At its heart, SNMP is a set of rules that allows a computer to get statistics from another computer across the Internet.

Computers keep track of various statistics that measure what they're doing. For example, routers can keep track of the number of bytes, packets, and errors that were transmitted and received on each interface (port). Web servers might keep a tally of the number of hits they have received. Other kinds of equipment has configuration information that's available through SNMP.

Each of these pieces of information (packet statistics, page hits, configuration) is kept in a database described by a *Management Information Base* (a *MIB* in SNMP parlance.) There are a many different MIBs, describing many different aspects of a computer's operation.

The various values that can be retrieved from a MIB are called *MIB variables*. These variables are defined in the MIB for a device. Each MIB variable is named by an *Object Identifier* (OID), which usually has a name in the form of numbers separated by periods ("."), like this: 1.3.6.1.xxxx.x.x.x.x...

For example, the MIB-II (pronounced, "MIB two") has a variable that indicates the number of interfaces (ports) in a router. It's called the "ifNumber", and its OID is 1.3.6.1.2.1.2.1.0

SNMP Watcher and InterMapper, as well as many other network monitoring tools, will query a device for the MIB variables and display the results. When a device receives a SNMP Get-Request for this ifNumber OID, it will respond with the count of interfaces.

Note: The trailing ".0" in the example above is technically part of the OID. Although you will often see OIDs written without it, InterMapper requires that it be present wherever you enter an OID.

What is the 'Read-only Community String'?

The SNMP Read-Only Community String is like a password. It is sent along with each SNMP Get-Request and allows (or denies) access to device. Most network vendors ship their equipment with a default password of "public". (This is the so-called "default public community string".) Many network administrators will change the community string to keep intruders from getting information about the network setup. This is a good idea. Even if it's only read-access, SNMP can divulge a lot of information about the network that could be used to compromise it.

If there's a "read only community string", you might expect that there is a "Write community string". You'd be correct. There is also a SNMP Set-Request, which is a command to set certain SNMP MIB variables (e.g., certain OIDs) to a specified value. These writes are protected by the write community string (which should *never* be set to 'public!'). Many SNMP-speaking devices also have IP address filters that ignore requests (read and write) unless the source address is on an access list.

There's also a SNMP Trap, which is an unsolicited message from a device to an SNMP console (for example, InterMapper) that the device is in an interesting state. Traps might indicate power-up or link-up/down conditions temperatures exceeding certain thresholds, high traffice, etc. Traps provide an immediate notification for an event that might only be discovered during occasional polling.

[Back to top](#)

Why can't I get SNMP information from a device?

InterMapper requires that SNMP be available and configured to display traffic information. The most common cause of not being able to see traffic is that you haven't entered the SNMP Read-only community string. (This is like a password that controls whether another computer can retrieve SNMP information.)

In order of simplest to most complex, here is a list of reasons that InterMapper might not get SNMP information from a device:

- Wrong DNS name/IP address (not likely, but we have to mention it)
- No connectivity. Can you ping the device from InterMapper?
- No SNMP agent on the device. Many devices or computers have optional SNMP capabilities that must be installed separately.
- Have you specified the OID properly? (See the [OID Format FAQ](#) for details.)
- Wrong Community string (have you tried 'public' ?)
- Access lists: does the equipment only allow SNMP access from certain addresses?
- Firewalls: does a firewall block the SNMP port between your Mac and the equipment?
- Bugs in the SNMP agent on the equipment. InterMapper uses SNMP Get-Next-Requests in several places. We've seen certain equipment that fails when queried this way.

If you're sure that you've checked all these things and you still can't get SNMP information, please get back to us at intermapper@dartware.com. We may have some tricks up our sleeves. (Or we may wind up learning something!)

[Back to top](#)

How can InterMapper query a particular MIB variable?

There are two kinds of MIB variables: scalar values and table entries. Scalars have a single value, such as the interface number shown above. For example, the ifNumber MIB variable of a router is a single number that represents the total number of its interfaces (ports). Table values, on the other hand, provide the same pieces of information for different items, such as the traffic for each of a router's ports, or information about each of the TCP connections in a device.

InterMapper can read and display both scalar variables and table variables in its custom SNMP probes.

Scalar values must have a ".0" suffix in their OIDs. For example, the OID for ifNumber in MIB-II is often written as "1.3.6.1.2.1.2.1". In custom probe files, it should be represented as "1.3.6.1.2.1.2.1.0". (This ".0" is technically part of the OID - it's convenient not to write it, though.)

Table variables are generally suffixed with the index of the row. (This isn't always true: see the note below). For example, the Cisco Environment Monitoring MIB defines two variables for the input air temperature and input voltage as the first rows in each of these tables:

```
ciscoEnvMonTemperatureStatusValue 1.3.6.1.4.1.9.9.13.1.3.1.3
```

```
ciscoEnvMonVoltageStatusValue 1.3.6.1.4.1.9.9.13.1.2.1.3
```

If you add a suffix ".1" to each of these, you'll get the value of the first row; add ".2" to as a suffix, you'll get the second row, etc.

Do all tables have an index?

As noted above, some tables don't have a separate index column. These rows are named (e.g., their OIDs are specified by) data in the row. For example, the OID for tcpConnState row, the status of a particular TCP connection is "1.3.6.1.2.1.6.13.1.1". Its index is the source and destination IP address and port (all four values) which are appended to the tcpConnState OID. Thus, the full OID for the state of a TCP connection from 9.8.7.6 port 543 to 123.45.67.89 port 8765 would be:

```
1.3.6.1.2.1.6.13.1.1.9.8.7.6.543.123.45.67.89.8765
```

[Back to top](#)

Where can I read more information about SNMP?

Here's a great site to start learning about MIBs and all the cool things you can do with them:

<http://www.snmpworld.com/>

Another is:

<http://netman.cit.buffalo.edu/>

A periodic newsletter, The Simple Times, is online at:

<http://www.simple-times.org/>

A great site pointing to various snmp products:

<http://www.simpleweb.org/>

[Back to top](#)

FAQ: Network and Server Probes

- [How do I change the protocol that a device is being polled with?](#)
- [What MIB variables does InterMapper poll?](#)
- [What SNMPv2c variables does InterMapper poll?](#)
- [How Does InterMapper Compute Traffic Statistics?](#)
- [How Does InterMapper Compute Utilization for a Link?](#)
- [How Does InterMapper Compute Errors for a Link?](#)

How do I change the protocol that InterMapper polls with?

Select the device and choose **Get Info...** from the **Network** menu. Use the pop-up menu at the bottom of the Get Info window to change the protocol. This is fully described in the [Device Information Window](#) section of the manual.

[Back to top](#)

What MIB variables does InterMapper poll?

InterMapper has two built-in SNMP probes: one for SNMPv1 devices, and one for devices that support the 64-bit counters of SNMPv2c.

For SNMPv1 devices, InterMapper examines the following MIB-II variables to determine the traffic flowing on a link.

MIB Variable	OID	Assoc. MIB
ifInOctets	1.3.6.1.2.1.2.2.1.10	MIB-II
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	MIB-II
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12	MIB-II
ifOutOctets	1.3.6.1.2.1.2.2.1.16	MIB-II
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	MIB-II
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18	MIB-II

InterMapper examines these two variables to decide whether an interface is up or down:

MIB Variable	OID	Assoc. MIB
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	MIB-II
ifOperStatus	1.3.6.1.2.1.2.2.1.8	MIB-II

InterMapper examines these variables to detect error conditions:

MIB Variable	OID	Assoc. MIB
ifInDiscards	1.3.6.1.2.1.2.2.1.13	MIB-II
ifInErrors	1.3.6.1.2.1.2.2.1.14	MIB-II
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	MIB-II
ifOutErrors	1.3.6.1.2.1.2.2.1.20	MIB-II

[Back to top](#)

What SNMPv2c variables does InterMapper poll?

InterMapper queries the following MIB variables in its SNMPv2c probe. The first set is used on an initial scan of the device; the second set is polled to display statistics for the device's operation.

MIB Variable	OID	Assoc. MIB
--------------	-----	------------

ifDescr	1.3.6.1.2.1.2.2.1.2	MIB-II
ifType	1.3.6.1.2.1.2.2.1.3	MIB-II
ifMTU	1.3.6.1.2.1.2.2.1.4	MIB-II
ifSpeed	1.3.6.1.2.1.2.2.1.5	MIB-II
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	MIB-II
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	MIB-II
ifOperStatus	1.3.6.1.2.1.2.2.1.8	MIB-II
ifName	1.3.6.1.2.1.31.1.1.1.1	RFC 2233
ifHighSpeed	1.3.6.1.2.1.31.1.1.1.15	RFC 2233
ifPromiscuousMode	1.3.6.1.2.1.31.1.1.1.16	RFC 2233
ifConnectorPresent	1.3.6.1.2.1.31.1.1.1.17	RFC 2233
ifAlias	1.3.6.1.2.1.31.1.1.1.18	RFC 2233

MIB Variable	OID	Assoc. MIB
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	MIB-II
ifOperStatus	1.3.6.1.2.1.2.2.1.8	MIB-II
ifLastChange	1.3.6.1.2.1.2.2.1.9	MIB-II
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	MIB-II
ifInErrors	1.3.6.1.2.1.2.2.1.14	MIB-II
ifInDiscards	1.3.6.1.2.1.2.2.1.13	MIB-II
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	MIB-II
ifOutErrors	1.3.6.1.2.1.2.2.1.20	MIB-II
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	MIB-II
sysUpTime	1.3.6.1.2.1.1.3	MIB-II
ifHCInOctets	1.3.6.1.2.1.31.1.1.1.6	RFC 2233
ifHCOutOctets	1.3.6.1.2.1.31.1.1.1.10	RFC 2233
ifInMulticastPkts	1.3.6.1.2.1.31.1.1.1.2	RFC 2233
ifInBroadcastPkts	1.3.6.1.2.1.31.1.1.1.3	RFC 2233
ifOutMulticastPkts	1.3.6.1.2.1.31.1.1.1.4	RFC 2233
ifOutBroadcastPkts	1.3.6.1.2.1.31.1.1.1.5	RFC 2233

Note: In the SNMPv2c probe, the input and output MulticastPkts and BroadcastPkts MIB variables replace NUcastPkts variables of the SNMPv1 probe, which are deprecated. HCOctets replace the regular Octets counters. Pkts and errors still use the MIB-II 32 bit counters.

[Back to top](#)

How Does InterMapper Compute Traffic Statistics?

InterMapper uses ifInOctets and ifOutOctets to compute the Receive and Transmit bytes/second values, respectively. The Receive and Transmit packets/second numbers are computed using the sum of the (ifInUcastPkts + ifInNUcastPkts) and (ifOutUcastPkts + ifOutNUcastPkts) respectively.

[Back to top](#)

How Does InterMapper Compute Utilization for a Link?

InterMapper queries a device at specified intervals, and requests a number of SNMP MIB variables. To compute utilization, InterMapper queries ifInOctets (and ifOutOctets) and the sysUpTime and ifSpeed variables. It then subtracts the octet counts from successive samples, and divides by the difference in the sysUpTime samples to compute a byte/second rate. It then divides this by the ifSpeed variable to compute a percentage of the link's capacity. (If the user has overridden the ifSpeed variable, InterMapper uses the user-entered value.)

If a network is using a shared baseband link (such as Ethernet, wireless, etc.) InterMapper compares the sum of the transmitted and received bytes/second against the link speed to get the utilization. If it's a full-duplex link (such as a frame relay, T-1 or T3, ATM, etc.) then InterMapper compares the higher of the transmitted or received data rate against the link speed.

[Back to top](#)

How Does InterMapper Compute Errors for a Link?

Q: A customer writes, "I see the Received Discards/Minute and Percent Err values for an ATM AAL5 interface are non-zero and I would like to know which variables and the calculation used for these numbers.

"We are also graphing the Percent Err: value. This figure is showing errors and my Cisco support folks wanted to know which MIB variables go into the calculation of this percentage and how they are combined to create this number."

A: The one-way percent errors under the Receive section are computed by totalling { ifInUcastPkts, ifInNUcastPkts, ifInErrors, ifInDiscards } as follows:

PERCENT ERROR = totalErrors / totalPkts;

where:

totalErrors = dErrs + dDis and
totalPkts = dUcast + dNUcast + totalErrors

and:

dUcast = ifCurrStats.inUcastPkts - ifPrevStats.inUcastPkts
dNUcast = ifCurrStats.inNUcastPkts - ifPrevStats.inNUcastPkts
dErrs = ifCurrStats.inErrors - ifPrevStats.inErrors
dDis = ifCurrStats.inDiscards - ifPrevStats.inDiscards

Note: Either of 'dErrs' or 'dDis' may be forced to zero if you have "Ignore Interface Errors" or "Ignore Interface Discards" checked.

The one-way percent errors for outgoing traffic are similarly computed from the { ifOutUcastPkts, ifOutNUcastPkts, ifOutErrors, ifOutDiscards } statistics.

The two-way Percent error number (just below Utilization on the Interface information pop-up) is the probability given both one-way error percentages that a packet will be lost making the round-trip across the link and back. If the probability of successful transmission is T and the probability of successful receipt is R (and assuming the act of transmission and receive are relatively independent), then the probability of a successful round-trip is T * R. The probability of error is (1 - T*R).

T and R are computed from the complement of the one-way percent errors above.

[Back to top](#)

FAQ: Miscellaneous

- [Do you have plans for a Windows or UNIX version?](#)
- [How do I launch helper applications?](#)
- [How do I resize the columns in the device list?](#)
- [If I look at the traffic on a link, wait five seconds, and look again, the traffic rates are the same. Shouldn't these numbers be updated?](#)
- [How does InterMapper compute byte and packet rates?](#)
- [How does InterMapper compute time intervals?](#)
- [Can InterMapper export data?](#)
- [Can InterMapper send data tagged with different MIME Types?](#)

[Back to top](#)

Do you have plans for a Windows or UNIX version?

We are currently thinking about what will be necessary to port InterMapper to other platforms.

[Back to top](#)

How do I launch helper applications?

Cmd-click on a device and InterMapper will display a popup menu that lets you launch a helper program. Helper applications are selected using Internet Config (or the Internet Control Panel) settings for Classic MacOS; InterMapper has pre-configured settings for MacOS X.

Internet Protocol (IP):

Ping MacPing <http://www.macping.com>

SNMP SNMP Watcher <http://www.snmpwatcher.com>

Telnet The configured Telnet application

Finger The configured Finger application

FTP The configured FTP application

HTTP The configured web browser

Traceroute The configured traceroute application

Timbuktu The Timbuktu client

AppleTalk Protocol:

Ping MacPing <http://www.macping.com>

SNMP SNMP Watcher <http://www.snmpwatcher.com>

MacOS X

Ping Launch the Terminal application with `/sbin/ping`

SNMP SNMP Watcher <http://www.snmpwatcher.com>

Telnet Launch the Terminal application

Finger The configured Finger application

FTP The configured FTP application

HTTP The configured web browser

Traceroute Launch the Terminal application with `/usr/sbin/traceroute`

Timbuktu The Timbuktu client

For Networks:

Launch MacPing to show all devices on that AppleTalk network or IP subnet.

[Back to top](#)

How do I resize the columns in the device list?

Option-click and drag the column labels.

[Back to top](#)

If I look at the traffic on a link, wait five seconds, and look again, the traffic rates are the same. Shouldn't these numbers be updated?

The traffic statistics are samples: the numbers will not change until after InterMapper probes the device again.

[Back to top](#)

How does InterMapper compute byte and packet rates?

SNMP only supplies counts of bytes, packets, or errors, etc. that have passed through or occurred in an interface. These counts increment "forever" (or until the counter rolls over to zero like a car's odometer).

During each poll, InterMapper collects the total traffic and computes the difference with the total traffic from the previous poll. It then divides by the amount of time that has passed to compute the rate (per second or per minute).

Technical note: Even when a counter rolls over (e.g., from 999 to 000), InterMapper will compute the traffic rates accurately. Let's say the two successive samples are 995 and 003. InterMapper subtracts the previous count (995) from the new count (003), and assuming that the "003" is actually "1003", and gets the proper difference of 8. Although the counters in the SNMP MIB variable are binary numbers, the same arithmetic principles hold. Thus InterMapper can compute these rates accurately.

[Back to top](#)

How does InterMapper compute time intervals?

To compute the elapsed time accurately, InterMapper uses the sysUpTime variable of the device as a timestamp to calculate the time that has elapsed between subsequent two polls. The time elapsed should roughly correspond to the poll interval; however, it is possible for polls to be delayed occasionally so using the change in sysUpTime to measure the elapsed time is more accurate.

[Back to top](#)

Can InterMapper export data?

InterMapper can export a map's contents in several formats. The simplest way to do this is to open the map, then choose **Export...** from the File menu. There are three formats available:

- A PICT file.
- An [EtherPeek](#) Names table. This is a tab-separated list of names and addresses with the following columns:

```
DNSname [tab] protocol [tab] MAC/IP address where available
```

- InterMapper can export information about a map in a tab-delimited file format, suitable for opening with Excel or other spreadsheet program. For devices, the columns are:

```
Status User-Defined Name Address Probe Type DNS/NBP Name sysUpTime sysName  
sysDescr sysContact sysLocation Comment User List
```

The column heads for links are:

Status Device Name ID ifDescr ifSpeed ifMtu Addresses ifLastChange PhysicalAddress

[Back to top](#)

Can InterMapper send data tagged with different MIME Types?

Yes. If there is a 'mimetypes' file at the top level of the *Web Pages* folder, it will define the correspondence between the template or target file's suffix and the MIME-type information that will be sent along with the file. Here is a sample mimetypes file:

```
# Sample MIMETypes file
# Format is: <file-suffix> <whitespace> <MIME-descriptor>

wml      text/vnd.wap.wml
wmls     text/vnd.wap.wmlscript

wbmp     image/vnd.wap.wbmp

wbxml    application/vnd.wap.wbxml
wmlc     application/vnd.wap.wmlc
wmlsc    application/vnd.wap.wmlscriptc
```

[Back to top](#)

FAQ: Open Transport SNMP

- [InterMapper crashes when I start it up...](#)
- [Other computers crash when I use InterMapper...](#)
- [How can I disable OpenTransport SNMP?](#)
- [Can I use OpenTransport SNMP with MacOS 9.1 or 9.2?](#)

InterMapper crashes when I start it up...

There is a known bug in the OpenTransport SNMP code that shipped with all versions of MacOS 9.0, and computers with AirPort software installed. Under certain circumstances, a Macintosh computer with MacOS 9.0.x and OpenTransport SNMP installed can crash when it receives an SNMP request packet. Because InterMapper sends SNMP queries in its normal operation, and it may cause a crash of the computer InterMapper is running on.

Dartware has reported this problem to Apple, and is working with them to come up with a fix.

This was a problem with versions before InterMapper 3.0.4, which tests the computer that it's running on, and quits if the OpenTransport SNMP software is installed.

Other computers crash when I use InterMapper...

The same bug that causes the crash above can affect remote Macintosh computers that have 9.0 and OpenTransport SNMP installed.

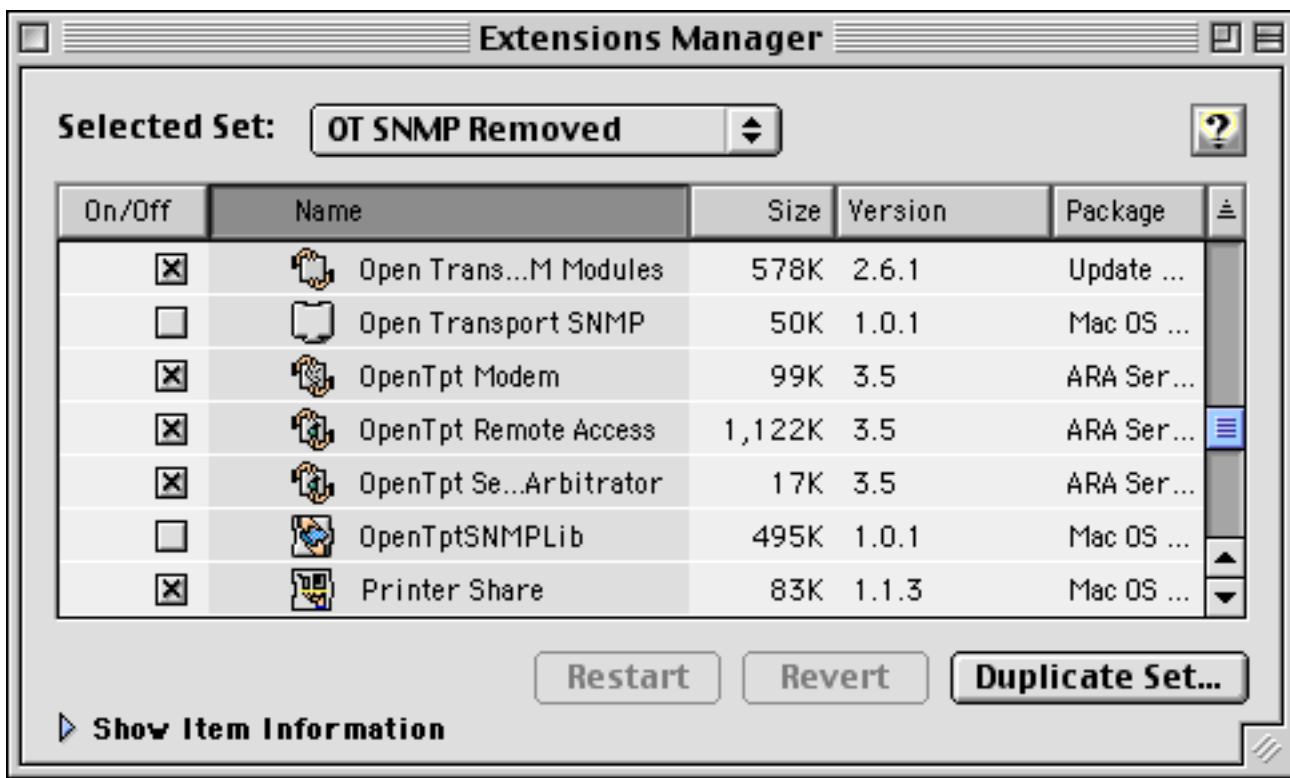
As a work-around, InterMapper (as of version 3.5 and later) now sends a single SNMP query to test for this combination on each computer it's testing with SNMP. If this combination is detected, InterMapper will place the device in Alarm and color its icon orange. Furthermore, it will no longer test that device with SNMP queries, but will ping the device instead. You can detect a Mac that has OpenTransport installed by clicking and holding on the icon. The "Reason" string in the popup window will display, "OpenTransport installed. May be cause of instability with MacOS 9."

Warning: Even though InterMapper will no longer cause that Macintosh computer to crash, other network monitoring tools that send SNMP queries may place that Mac at risk. To avoid this, you should disable the OpenTransport SNMP software on all computers where it is not essential. (And for those, you must evaluate whether the risk of crashes is worth the data that can be retrieved from the machine through SNMP.) The process is described below:

How can I disable OpenTransport SNMP?

To disable OpenTransport in MacOS 9, use the Extensions Manager to disable these two files listed below and then restart the computer, as shown in the figure below.

- Open Transport SNMP
- OpenTptSNMPLib



Can I use OpenTransport SNMP with MacOS 9.1 or 9.2?

No. As of this writing (Thu, Jan 24, 2002), neither of the installers for MacOS 9.1 or 9.2 includes the capability of installing OpenTransport SNMP. If you attempt to install the version from 9.0, you will run into the problems described above.

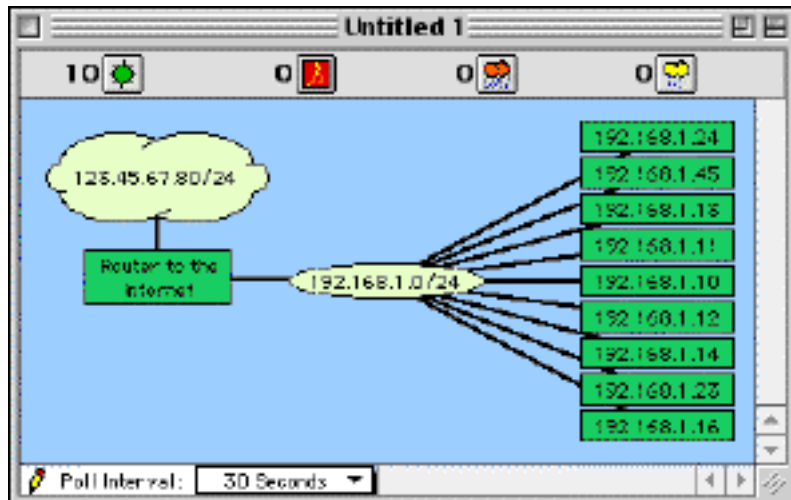
Adding Unmanaged Hubs and Switches to a Map

InterMapper cannot automatically discover or monitor unmanaged switches and hubs (so-called 'dumb' devices) since they have no IP address. However, there is a workaround that allows you to display them on an InterMapper map.

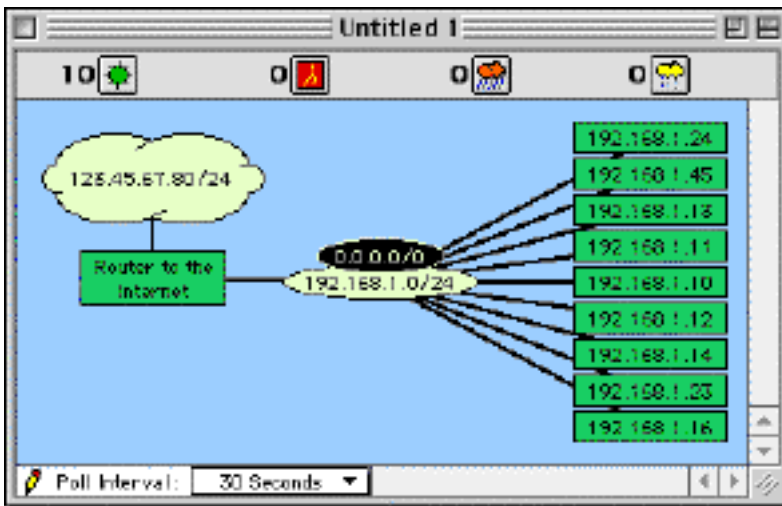
To do this, you can create a placeholder icon, and then manually drag the links from the appropriate devices to this new icon. Although InterMapper cannot test this "fake" equipment, it will appear on the map and display the interconnections of your network as a tool to diagnose problems.

Here is a step-by-step description of the process. Note that this description works equally well for either switches or hubs.

In the starting map, notice that InterMapper has automatically connected a number of devices to the network oval labeled "192.168.1.0/24". We happen to know that the top three devices--IP addresses 192.168.1.24, .45, and .13--are in fact, connected to a dumb (e.g., unmanaged) hub on the floor above. This page shows how to create a placeholder icon to represent the hub and connect those three devices to it.

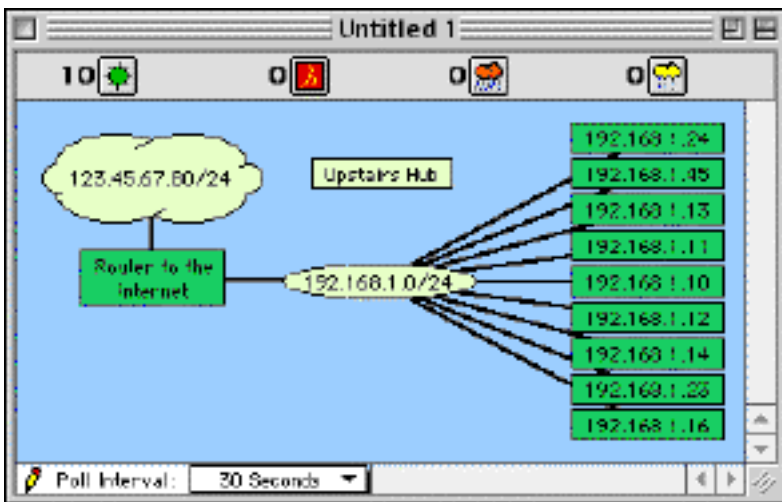


1. The top three devices - IP addresses 192.168.1.24, .45, and .13 - are in fact, connected to a dumb (e.g., unmanaged) hub upstairs. We want to create a placeholder icon that represents the hub, and then move the connections for those devices to the placeholder.



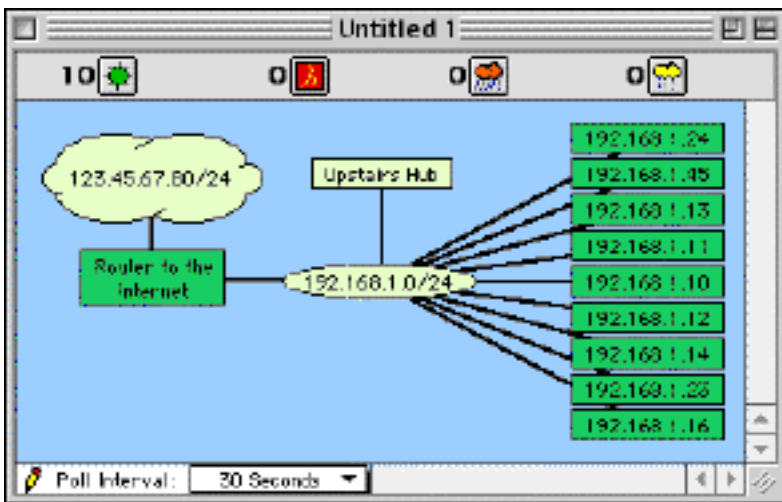
2. The first step is to create a new (empty) network. To do this, choose **Add Network...** from the **Network** menu. Enter a subnet number that's the same as the device's current subnet (oval) as shown in Figure 1-23 of the [Adding Subnets](#) page.

You should see the new network appear as an oval with the subnet number you entered (not "0.0.0.0/0", as shown in this example).



3. We can tidy up the appearance of the item by doing the following:

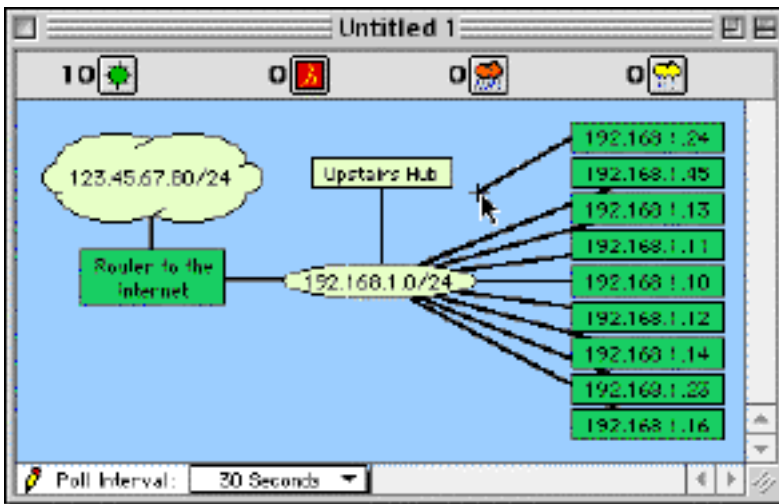
- Move it up a little bit
- Change its shape to a rectangle using the **Shape** command in the **Display** menu
- Change its name to "Upstairs Hub" using **Label (Cmd-L)**



4. Next connect this new rectangle to the oval below. To do this:

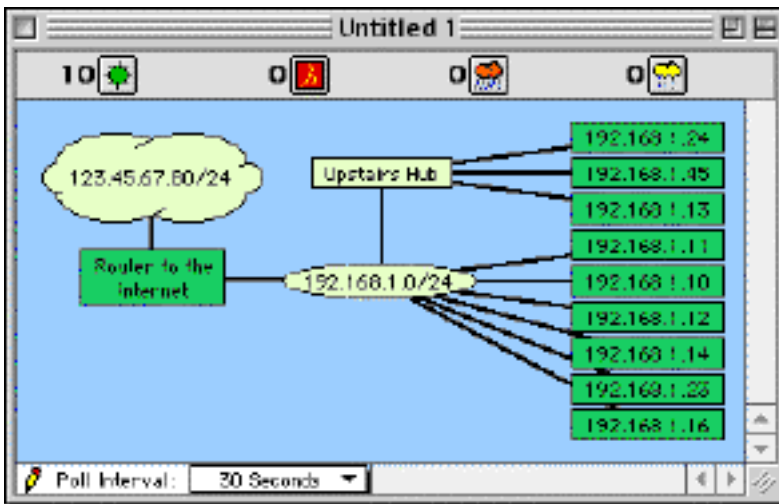
- Hold down the "e" key
- Drag the cursor from one item to another.

You should see that a line has connected them together. This line will persist as you move the items around a map.



5. Now drag each of the links for the three devices from the oval to the new rectangle. To do this, click on a link (line) and drag it toward the "Upstairs Hub" rectangle. Let go of the mouse when it's over the rectangle.

The line will stick to the new rectangle. Do this for all three links.



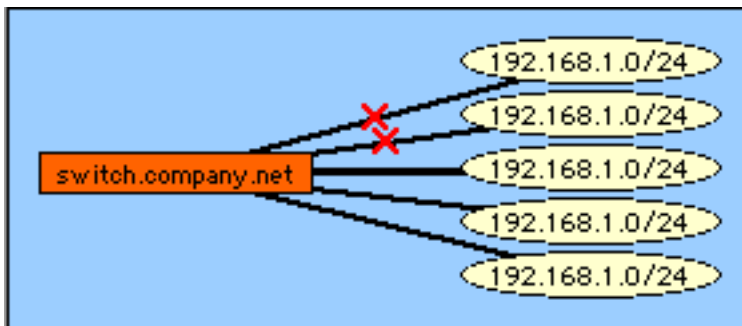
6. The final image, after dragging the three links from the oval to the "fake" hub.

Note that the rectangle is yellow: InterMapper cannot monitor it because this rectangle was created as a placeholder -- it isn't a real device. However, it is convenient to have the hub represented on the map to indicate the actual interconnections between the devices.

Changing Labels on Switch Ports

InterMapper shows the subnet information (e.g., the IP address and subnet mask) for ports on managed switches and hubs. This information can clutter the screen, and may not always be useful.

Here is a procedure for changing the port labels to contain their port number instead of the default network/subnet information.

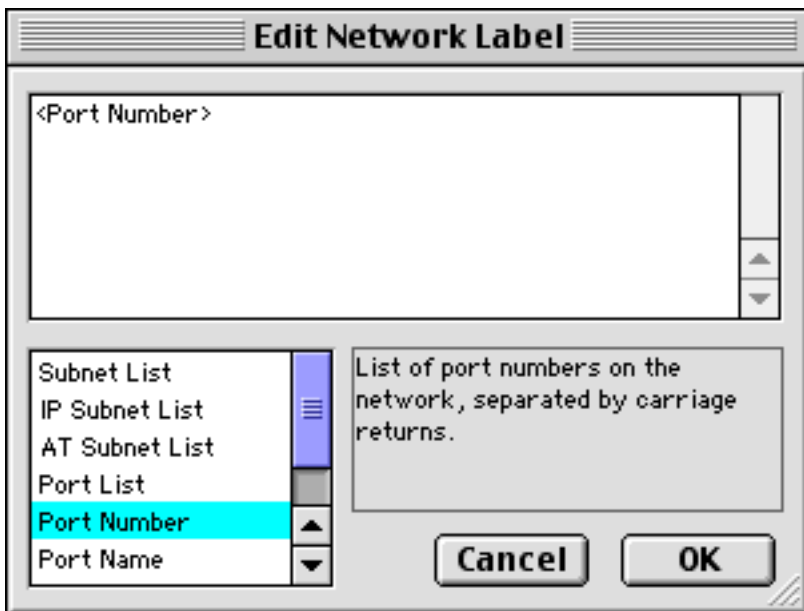


1. The starting map. Select all ports by shift-clicking or dragging across the ports, or click the switch then choose **Select Adjacent** (Cmd-J) from the **Edit** menu.



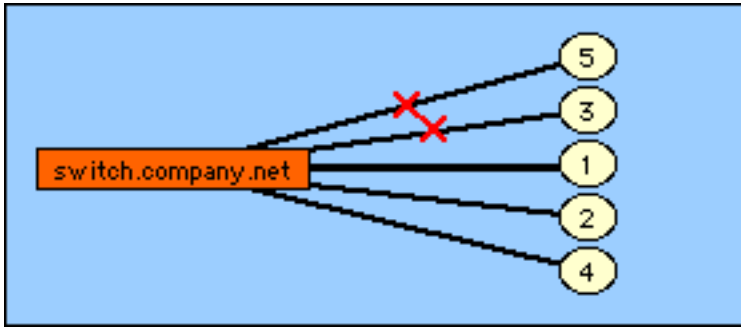
2. Choose **Label...** (Cmd-L) from the **Display** menu.

If devices and networks are both selected, you will see the warning shown at the left. Click **Networks** to modify the networks' labels.



3. This window allows you to change the labels of network ovals. The text in the top pane of this window will become the label for the selected item(s). Variable names (which are bracketed in "< ... >") will be replaced with their actual value when the labels are displayed.

To add the port number to the label, double-click "<Port Number>" in the list at the lower left. You'll see that it is inserted into the top pane. Click **OK**.



4. This is the final map. Notice that the network ovals are now labeled with the port number.

FAQ: Using InterMapper on MacOS X

- [What differences are there between the Classic and MacOS X versions?](#)
- [Why must I grant network permissions to InterMapper when it starts?](#)
- [Why do I get a "Read only media" message when I launch InterMapper?](#)
- [InterMapper won't run from a local hard drive](#)

[Back to top](#)

What differences are there between the Classic and MacOS X versions?

A: We have striven to keep the Classic MacOS version (7.5 to 9.2) and the MacOS X versions of InterMapper nearly identical. There are a small number of changes though:

- Preference files are stored in different folders/directories. See [Appendix D -- Files and Folders](#)
- If you wish to backup the files on MacOS X, be sure to use Stuffit. See [Appendix D -- Making Backups](#)
- The MacOS X version (in version 3.6) will use OpenSSH code for the HTTPS probes; the Mac Classic version uses the built-in URL Access facility.

[Back to top](#)

Why must I grant network permissions to InterMapper when it starts?

A: InterMapper must setuid to become an Administrator ("root") in order to open the Ping port, the SNMP port, and certain other low-numbered ports (such as port 80 for the web server). It must be authorized (a single time) by a system administrator to do this. For details, you can read the [OS X Permissions page](#).

[Back to top](#)

Why do I get a "Read-only file system" message when I launch InterMapper?

Q: When I double-click InterMapper and try to Grant Network Permissions, I get an error message that states, "Permission could not be granted because an error occurred. (-1) Read-only file system". What does this mean?

A: You probably double-clicked the copy of InterMapper on the disk image (.dmg) volume. You should drag the InterMapper folder to your hard disk, then double-click it. InterMapper must change the permissions on its own binary file. To do this, it must be saved on a writable volume; the disk image is read-only.

[Back to top](#)

InterMapper won't run from a local hard drive

Q: I've copied InterMapper from the disk image (.dmg) to a local (but non-boot) hard drive. But it still won't run. Any ideas?

A: The most likely cause is that the access bits on the InterMapper application aren't being set properly. There are a couple ways to check this:

1) If you're handy with the Terminal application, you can:

1. Open the Terminal.app program (in the Applications/Utility folder)
2. cd to the InterMapper folder then cd into the InterMapper "package" (It's simpler if remove the space and registered trademark ((R)) symbol from the folder and file names.)

The InterMapper application is in:

InterMapper:Contents:MacOS:InterMapper

Keep cd'ing until you get into the MacOS folder, then type `ls -al`

You should see something like this:

```
total 6904
drwxrwxrwx  3 richb  staff      58 Oct 10 11:39 .
drwxrwxrwx  6 richb  staff     160 Oct 10 11:39 ..
-rwsr-xr-x  1 root   staff 3531952 Oct 10 11:39 InterMapper
```

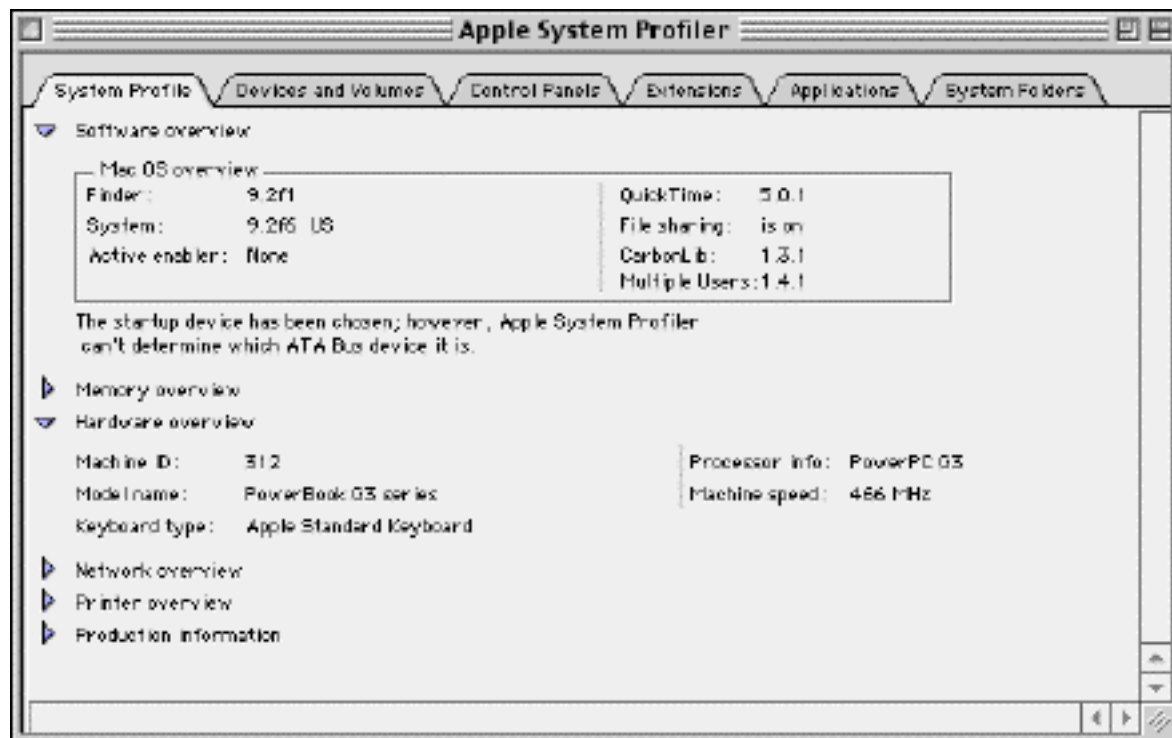
If the file permissions have been set properly, the leftmost four characters should be -rws, and the file should be owned by "root" (in the third column).

2) If you'd prefer a graphical tool, grab a copy of xFiles from http://personalpages.tds.net/~brian_hill/downloads.html. It's an application that will show you file permissions, and it's good for changing many files' permissions at once. To use it:

1. Open xFiles. You'll see a window like the one shown in its web page.
2. Control-Click on the InterMapper icon, and select "Show Package Contents" from the resulting contextual menu. You'll see a new window with a "Contents" folder.
3. Open the Contents folder, and then the enclosed "MacOS" folder. You'll see an InterMapper icon (it's just a plain icon.)
4. Drag the InterMapper icon to the xFiles window. I always drop it onto the area near the checkboxes.
5. The window will show InterMapper's attributes. The important ones are:
6. In the Owner column of checkboxes, the Read and Execute should be checked.
7. The Owner (at the left) should be "root"
8. The Posix attributes (click the right triangle at the right) should have only the Set User ID box checked.

[Back to top](#)

Creating an Apple System Profiler Report

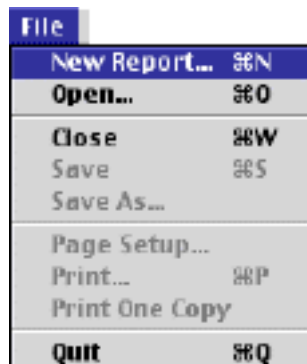


The **Apple System Profiler** is a tool for automatically collecting information about Macintosh computers. It is available for both Classic MacOS (9.0 and above) and MacOS X. To create a report:

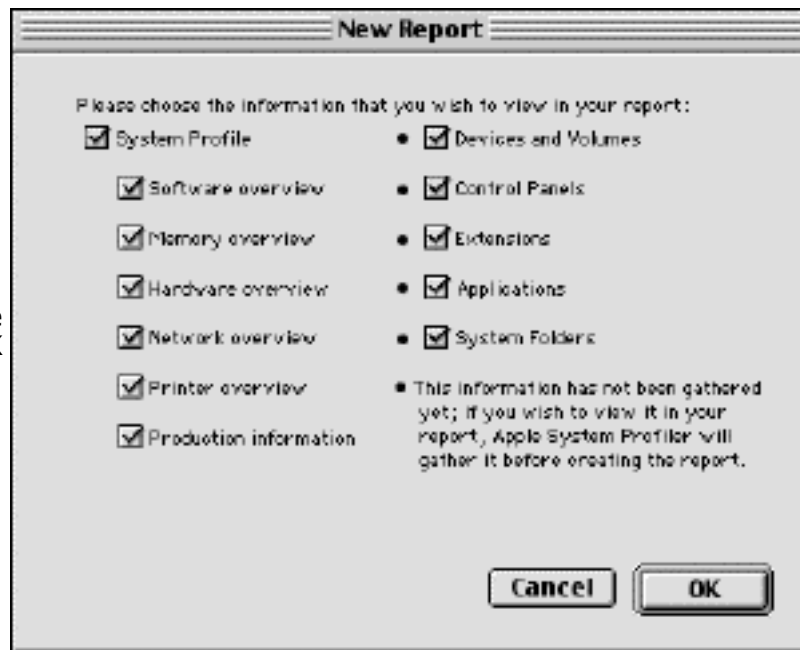
- **MacOS Classic:** Open the Apple System Profiler from the Apple menu.
- **MacOS X:** Open the Apple System Profiler from the /Applications/Utilities folder.

In either case, you'll see a window similar to the one at the top of the page.

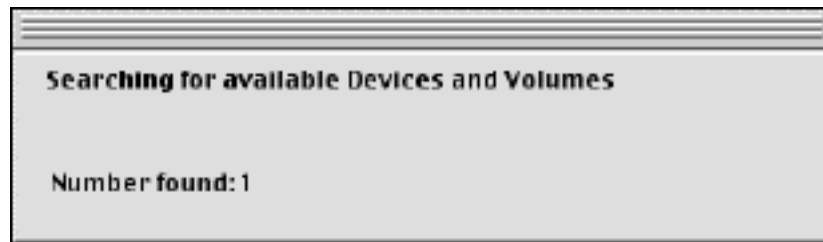
Choose **New Report...** from the **File** menu



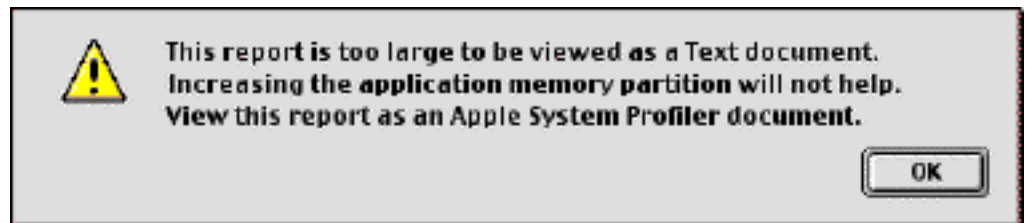
Check all the boxes in the resulting window and click **OK**



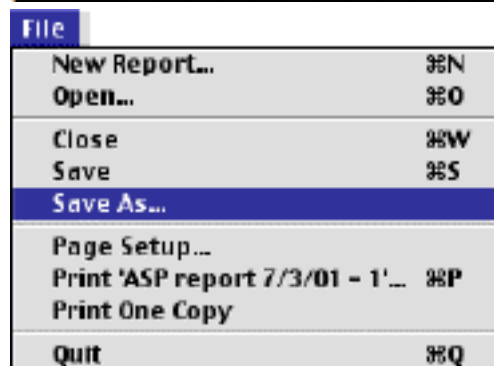
It may take a few moments to accumulate all the desired data for the report



You may also see this window. Just click **OK**



Save the file (choose **Save** from the File menu)



FAQ: What do InterMapper's Error Codes Mean?

- [Error -108 - Out of memory](#)
- [Error -2102 and Problems with Paging](#)
- [Error -5000 when saving a file](#)
- [Error -50 when saving a file](#)
- [Error Type 25 when starting up](#)
- [Error: 1008:17,-23 -- MVTS#12666# during Installation](#)
- [Errors -6986 and -6996 with HTTPS servers](#)
- [Port In Use error while paging](#)
- [Error -3247 opening UDP Socket](#)

InterMapper attempts to provide human-readable explanations for all the problems that occur. Occasionally, an error condition will arise for which InterMapper doesn't have a text string. In these cases, you'll see a numeric error code. This page lists the codes we've seen and the reasons/fixes for them.

Note: When InterMapper reports any error, it also logs the source code location of that error in the Debug window, described in [Chapter 5](#). Copy and paste the bottom 30 or so lines and e-mail them to us at support@dartware.com.

Error -108 - Out of memory

In the Classic MacOS, each application must be allocated a certain amount of RAM for its functions. To set this allocation, use the Finder's Get Info window. InterMapper defaults to 5 MBytes (5120 KBytes) which is enough for a few maps. For more or larger maps, you should allocate more memory to InterMapper.

Review the memory statistics in the **About...** box to see how much memory you currently use; we recommend that you run below 70% of the allocated memory.

[Back to top](#)

Error -2102: Problems with Paging

PageNOW! Personal Edition 2.0 will allow other applications (such as InterMapper) to page a single subscriber designated as the "default subscriber" under preferences. The PageNOW! Server software allows InterMapper to page multiple people. The -2012 error code indicates that the pager software cannot page the individual you specified. This is most likely caused because you are using the Personal Edition version of the software. You can upgrade the PageNOW! software from [our order page](#) or directly from [Mark/Space Softworks](#).

[Back to top](#)

Error -5000: Can't save a file

This is usually a file permission problem. There are two possible causes:

1) On MacOS X/Darwin, if a file is owned by the account XYZ, and root doesn't have group or world write access to it, root gets errors accessing the file. (This is different from Linux, where root access ignores all file permissions.) If that's the case, you can "chgrp wheel map; chmod g+w map" where map is the actual filename.

2) Also, the file may be locked. To see the file's locked status, use "ls -lo". You can unlock files with chflags (man chflags).

[Back to top](#)

Error -50: Can't save a file

We have heard of problems with InterMapper reporting a -50 error when attempting to save a file to a UFS (Unix File System) hard drive, instead of the default HFS+ (Hierarchical File System). We are investigating this. Saving to a HFS+ file system will likely work around this until we can release an updated version of InterMapper.

[Back to top](#)

Error Type 25 when starting up

InterMapper (Classic) may give a Type 25 error when starting up. This has been caused by not having enough memory allocated to the InterMapper application to open all the maps. To fix this, allocate more memory to InterMapper by Getting Info on the application, selecting Memory from the popup window, and setting the Preferred value larger.

[Back to top](#)

Error: 1008:17,-23 -- MVTs#12666# during Installation

When installing InterMapper (Classic), we have some reports that the error message above will appear. Conversations with our Installer vendor (MindVision) indicate that this may be related to AOL's Instant Messenger. They recommend that you use the Extensions Manager control panel to switch to MacOS Base extensions, run the installer, then switch the extensions back on.

If this does not immediately work, please send an e-mail to support@dartware.com and we will work with you to resolve this problem.

[Back to top](#)

Port In Use error while paging

The FAQ at the Mark/Space web site - <http://www.markspace.com/pagenow/faq.html#supportfaq> - covers many reasons there could be a "port in use" error message. In particular, you should also check for the dialup feature of Timbuktu Pro.

To disable Timbuktu's dialup functionality without interfering with its network code, locate the "Dial Direct Dropin" file in the System Folder:Extensions:Netopia" folder. Drag this file to the Trash (or move to a safe place outside the system folder) and then restart your system.

[Back to top](#)

Error -3247 opening UDP Socket

InterMapper may give this error message on MacOS X when there is a conflict with a different application for use of a port.

[Back to top](#)

Errors -6986 and -6996 with HTTPS servers

On Classic MacOS, InterMapper 3.6 and earlier uses the built-in URL Access facility to test HTTPS servers. This works fine as long as the server uses certificates derived from well-known Certificate Authorities. If your HTTPS server uses a private certificate, InterMapper will show the server in alarm with a "-6986" or a "-6996" error. InterMapper running on MacOS X is not subject to this problem.

[Back to top](#)

New Frequently Asked Questions

This page lists newly answered questions that will be moved to other pages in time. Please come back from time to time to see what's new!

I need to change the person who's paged on a monthly basis...

Q: We use InterMapper to page one of three on-call staff. There are three people in the on-call rotation and we have two maps, network map and servers. There are three subscribers configured that point to the cell phones (using PageNOW!) of the three on-call people. Is there a way to change all the assignments?

A: Yes, it's fairly straightforward. Here's the procedure:

1. You need to set up your subscribers in the PageNOW software. Call them "On Call 1", "On Call 2", and "On Call 3" (or Alice, Bob, Carol, or whatever you like).
2. You'll also need to create names in the Notification List (Cmd-U) that are the individuals responsible for your various devices. You might name them "Server Admin" and "Network Admin" (or however you have the responsibilities broken out). Set them up to page the proper individual (for today's schedule).
3. You'll need to assign devices to the appropriate name in the Notification List. The easy way to do this is to open one of the names from that list and click the "Show Responsibilities" button. You'll see a list of all the devices: check the up/down/etc. boxes for each of machines that person's responsible for.

Tip: You can option-click on a checkbox to check/uncheck an entire column of boxes.

4. Each time the on-call duties change, simply go to the Notification List, and re-set the pager to notify for the Server Admin and/or Network Admin.

[Back to top](#)

Is there a way I can submit bug reports for the MacOS X version?

This has been moved to [Chapter 4 -- Debugging Preferences](#).

[Back to top](#)

It takes a long time to do auto discovery on my network, 10.200.0.0/16. Why?

The subnet you describe is a private, Class A subnet with a 16 bit subnet mask (255.255.0.0) that includes 65,534 separate IP addresses. InterMapper limits its autodiscovery probes to two per second (so that it doesn't overload any networks) and thus it will take about 32,000 seconds (a shade under 10 hours) to scan that subnetwork completely.

To create your maps more quickly, you might want to type or paste one or more host DNS names or IP addresses into the Add Devices... window (Network menu). InterMapper will immediately add them to the map and connect them to the proper network.

[Back to top](#)

In the Get Info window, there's a "Timeout (in seconds):" field. What does this control?

This is the timer that packet-based probes use to determine when a device has failed to respond. InterMapper sends the proper probe packet and if the device fails to respond within that time, InterMapper tries again two more times. If no response comes back, then InterMapper indicates the device is down. The timeout is operational for Ping/Echo, SNMP, BlitzWatch, DHCP/BootP, DNS, KeyServer, NTP, RADIUS, and

RTMP probes.

This timeout is not operational for the connection (TCP) based probes. Those probes use two timers. When a TCP probe initiates the connection, it waits up to 60 seconds for the connection to complete. A TCP probe uses a separate timer, defined within the script - usually 30 seconds - to wait for responses from the device.

[Back to top](#)

My maps get incredibly messy. The interfaces (along with the little ovals) on my switches keep appearing, even after I delete them. How can I keep this from happening?

InterMapper periodically scans routers and switches and displays newly discovered interfaces. If you delete the interface/oval from the map, InterMapper will rediscover it and display it again.

To keep the interfaces from showing up, simply select them and choose Hide from the Edit menu. InterMapper will not display them, yet they'll be in InterMapper's internal tables, and thus won't be rediscovered.

[Back to top](#)

How can I find out how many services I'm monitoring with InterMapper. Do I have to count all the boxes on each map?

Look in the About... box of InterMapper. The first pane shows the number of "Monitored services".

[Back to top](#)

Can InterMapper handle large networks? How big? What are the limits?

There's no hard and fast rule. A modest network with a few maps and a hundred or so devices will run nicely in 16 MBytes of RAM on pretty much any Macintosh: a PowerPC machine isn't even necessary. At Dartware, we test InterMapper on a Quadra 700 with 20 MBytes RAM.

The largest collection of maps on one single InterMapper that we know of is 56 maps, with over 1050 devices (routers, switches, servers, etc.) That customer runs InterMapper on a Macintosh G3 and allocates 40 MBytes of RAM to the program. (InterMapper's About... box indicates that it's using under 10 MBytes of RAM, though.)

[Back to top](#)

In my strip charts, I'm only able to keep 24 hours of data. I'd like to keep more like a week's data. Can I increase this?

InterMapper strip charts save 1000 data points by default. Each time the map polls the device (controlled by the Poll Interval popup at the lower left corner of the map), it adds a sample to the strip chart. When the limit is reached, InterMapper discards the oldest reading.

For example, if your map is polling devices once per minute, there will be about 1500 samples generated per day ($24 * 60 = 1440$ samples). To keep a week's worth of data, sampled every minute, you should have 10,080 data points per chart.

You can change the number of data points kept by strip charts. To do this, open the Chart Defaults category from the Preferences window (in the Edit menu). Change the number of points saved.

Increasing the number of data points will increase the amount of memory that InterMapper uses. InterMapper uses eight bytes per sample. An increase of 1000 data points increases the amount of memory by 8 Kbytes *per chart*. If you have 10 charts open, InterMapper will consume 80 KBytes more memory. Check the About... box to see how much memory is still available to the application. If it is running short, you

can increase InterMapper's allocation using the Finder's Get Info... window.

[Back to top](#)

My switches and hubs don't speak SNMP. Can I monitor traffic through them?

No. InterMapper queries the MIB of network equipment using SNMP to compute and display the traffic being processed by each interface.

My Network Administrator won't tell me the SNMP Read-only community string for the network equipment in our department. Can I get statistics for the links? Can my map show them with the devices that are connected?

You won't be able to see traffic statistics. The SNMP Read Only community string is like a password. If you don't use the correct string when making SNMP queries, the device will ignore the request. (You might try the industry-standard default community string of "public".)

You can, though, manually create the interconnections as described in the question about dumb hubs, above.

[Back to top](#)

What's the best way to scan my AppleTalk network and add the devices to my map?

You can add devices manually, using the Add Devices... command from the Network menu. You can also use the MacPing program (<http://www.macping.com>) to scan the network for AppleTalk devices. You can then copy the devices in MacPing and paste them into the InterMapper map.

[Back to top](#)

I use my switch's VLAN capabilities to segment my network. Is there a good way to show which equipment is connected to each VLAN segment?

Currently, you must manually drag devices to the proper port to indicate the correct connection point.

[Back to top](#)

How do delayed notifications work?

InterMapper maintains a queue of notifications to be sent. When a DOWN, WARN, or ALARM event happens, InterMapper places a notification in the queue, and sets its "time to be sent" according to the delay. (UP, OK and Trap notifications are never delayed.)

When an UP or OK event occurs, InterMapper first searches the notification queue for the corresponding down, warn, or alarm notification. If it's there, InterMapper removes both the DOWN (or Warn or Alarm) notification and UP (OK) event and won't send either one. If not, then InterMapper sends the UP/OK notification straight away.

[Back to top](#)

I can't test my SNAP Server with AppleShareIP. What's going on?

You might want check the firmware version of the SNAP Server. InterMapper could not test version 2.3 of the SNAP server with an AppleShareIP probe. (It received an immediate disconnect with a Reason code of

"Received disconnect after 0 seconds".) As of this writing (Mon, Nov 5, 2001), the current version of the SNAP Server OS is 3.1.618. Updates may be downloaded from: <http://www.snapserver.com>.

[Back to top](#)

Is it possible to get network traffic statistics from SNAP Server via SNMP?

The latest version of the SNAP Server OS supports SNMP. See the URL above for getting an update to the firmware.

[Back to top](#)