

Propel Administrator's Guide

Document Number 007-3272-001

CONTRIBUTORS

Written by Jeffrey B. Zurschmeide
Engineering Contributions by John Schimmel, Balaji G.
Document Production by Michael Dixon

St. Peter's Basilica image courtesy of of ENEL SpA and InfoByte SpA. Disk Thrower image courtesy of Xavier Berenguer, Animatica

© Copyright 1995, 1996, 1997 Silicon Graphics, Inc.— All Rights Reserved.

The contents of this document may not be disclosed to copied or duplicated in any form, in whole or in part, without the prior written permission of Silicon Graphics, Inc.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure of the technical data contained in this document by the Government is subject to restrictions as set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013 and/or in similar or successor clauses in the FAR, or in the DOD or NASA FAR Supplement. Unpublished rights reserved under the Copyright Laws of the United States. Contractor/manufacturer is Silicon Graphics, Inc., 2011 N. Shoreline Blvd., Mountain View, CA 94043-1389.

Silicon Graphics and IRIS are registered trademarks and IRIX and IRIS InSight are trademarks of Silicon Graphics, Inc. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

Contents

List of Figures	vii
List of Tables	ix
About This Guide	xi
What This Guide Contains	xi
How to Use This Guide	xi
Target Audience of This Guide	xi
Additional Resources	xii
IRIX Admin Manual Set	xii
Reference Pages	xiii
Release Notes	xiv
The IRIX Help System	xiv
The Silicon Graphics World Wide Web Site	xv
Conventions Used in This Guide	xv
User Interface Terminology	xvi
Common User Interface Operations	xviii
Using Scroll Bars	xviii
Entering and Removing Text in a Field	xix
Using Option Buttons	xx
Using a File Prompter	xx
Using Online Help	xxi
1. Propel Administration	1
propel Installation Instructions	2
The propel Databases	2
The Host Database	3
The Collection Database	4

- Initializing the Host Database 5
- Automatically Loading the Host Database 5
- Creating the Collection Database 8
- Populating the User and Group Databases 8
- Loading the File Database 8
- The propel License Manager (lm_tcp) 9
 - propel and rdist 9
 - Server/Client Security 10
 - Running snmpd on Your Network 10
- How to Use propel 11
 - The propel Management System 11
 - How propel Works 13
 - Using propel to Manage a Domain 15
 - Distributing Silicon Graphics Software With propel 15
 - Internal Data Distribution With propel 15
 - Installing Free Software Packages With propel 16
 - Host Information Management With propel 16
 - User Information Management With propel 16
 - Localizing propel 17
- Using the propel Databases 17
 - The Host Database 17
 - Editing the Host Database 19
 - Inserting a Host Entry 21
 - Updating a Host Entry 21
 - Querying the Host Database 21
 - Parsing System Files Into the Host Database 22
 - The Collection Database 22
 - Editing the Collection Database 22
 - Adding a Collection of Hosts to the Collection Database 23
 - Updating a Collection Database Entry 29

The File Database	29
Editing the File Database	30
Adding a File to the File Database	31
Updating a File Database Entry	33
Querying the File Database	33
The Distribution Rule Database	33
Editing the Rule Database	34
Adding a Distribution Rule to the Rule Database	34
Updating a Rule Database Entry	36
Querying the Rule Database	36
The User Database	36
Editing the User Database	37
Adding a User to the User Database	39
Updating a User Entry	39
Querying the User Database	39
The Group Database	40
Editing the Group Database	40
Adding a Group of Users to the Group Database	42
Updating a Group Database Entry	46
Additional Database Query Techniques	47
Query Keywords and Glob Expressions	49
Query Shortcuts	51
Using the dbredit Utility	52
Running propel	54
Running propel From a Shell Command	55
Running propel Through cron(1M)	55
The propel File Generation System	55
Index	57

List of Figures

Figure i	Window Terms	xvi
Figure ii	More Window Terms	xvii
Figure iii	A Horizontal Scroll Bar	xix
Figure iv	An Entry Field	xix
Figure v	An Option Button and an Option Button Menu	xx
Figure vi	A File Prompter Window	xxi
Figure vii	A Help Menu and a Help Button	xxi
Figure 1-1	The propel Directory View	12
Figure 1-2	The Host Database Editor	20
Figure 1-3	The Collection Manager	23
Figure 1-4	The Add Collection Window	24
Figure 1-5	A Populated Collection Manager Window	25
Figure 1-6	A Collection Display Window	26
Figure 1-7	The Add Host Window	27
Figure 1-8	The Add Host Query Window	27
Figure 1-9	A Populated Collection	28
Figure 1-10	The File Database Editor	31
Figure 1-11	A List of Source Files in the File database	32
Figure 1-12	The Distribution Rules Editor	34
Figure 1-13	A Prepared Distribution Rule	35
Figure 1-14	The User Database Editor	38
Figure 1-15	The Group Manager	41
Figure 1-16	The Add Group Window	43
Figure 1-17	A Populated Group Manager Window	44
Figure 1-18	A Group Display Window	45
Figure 1-19	The Add User Window	46

List of Tables

Table i	Outline of Reference Page Organization	xiv
----------------	--	-----

About This Guide

“About This Guide” includes brief descriptions of the contents of this guide and an explanation of typographical conventions used, and refers you to additional sources of information you might find helpful.

This guide explains how to perform general operation tasks of the *propel* software distribution and host and user management system, used with Silicon Graphics® workstations and servers.

What This Guide Contains

This guide contains the following chapter:

Chapter 1, “Propel Administration” provides an overview of *propel* and its uses.

How to Use This Guide

This guide is written for administrators who are responsible for configuring and maintaining the *propel* implementation at their site.

Target Audience of This Guide

This guide is intended for administrators who are responsible for one or more systems running the *propel* software distribution and distributed system management application.

Additional Resources

For easy reference, here is a list of the guides and resources provided with your system and the specific focus and scope of each:

IRIX Admin Manual Set

This guide is an additional resource to the *IRIX Admin* manual set.

The *IRIX Admin* suite is intended for administrators: those who are responsible for servers, multiple systems, and file structures outside the user's home directory and immediate working directories. If you find yourself in the position of maintaining systems for others or if you require more information about IRIX™ than is in the end-user manuals, these guides are for you. The *IRIX Admin* guides are available through the IRIS InSight™ online viewing system. The set comprises these volumes:

- *IRIX Admin: Software Installation and Licensing*—Explains how to install and license software that runs under IRIX™, the Silicon Graphics implementation of the UNIX® operating system. Contains instructions for performing miniroot and live installations using Inst, the command line interface to the IRIX installation utility. Identifies the licensing products that control access to restricted applications running under IRIX and refers readers to licensing product documentation.
- *IRIX Admin: System Configuration and Operation*—Lists good general system administration practices and describes system administration tasks, including configuring the operating system; managing user accounts, user processes, and disk resources; interacting with the system while in the PROM monitor; and tuning system performance.
- *IRIX Admin: Disks and Filesystems*—Describes how to add, maintain, and use disks and filesystems. Discusses how they work, their organization, and how to optimize their performance.
- *IRIX Admin: Networking and Mail*—Describes how to plan, set up, use, and maintain the networking and mail systems, including discussions of sendmail, UUCP, SLIP, and PPP.
- *IRIX Admin: Backup, Security, and Accounting*—Describes how to back up and restore files, how to protect your system's and network's security, and how to track system usage on a per-user basis.

- *IRIX Admin: Peripheral Devices*—Describes how to set up and maintain the software for peripheral devices such as terminals, modems, printers, and CD-ROM and tape drives. Also includes specifications for the associated cables for these devices.
- *IRIX Admin: Selected Reference Pages* (not available in InSight)—Provides concise reference page (manual page) information on the use of commands that may be needed while the system is down. Generally, each reference page covers one command, although some reference pages cover several closely related commands. Reference pages are available online through the `man(1)` command.

Reference Pages

The IRIX reference pages (often called “man” or “manual” pages) provide concise reference information on the use of IRIX commands, subroutines, and other elements that make up the IRIX operating system. This collection of entries is one of the most important references for an administrator. Generally, each reference page covers one command, although some reference pages cover several closely related commands.

The IRIX reference pages are available online through the `man` command. To view a reference page, use the `man` command at the shell prompt. For example, to see the reference page for `diff`, enter:

```
man diff
```

It is a good practice to print those reference pages you consistently use for reference and those you are likely to need before major administrative operations and keep them in a notebook of some kind.

Each command, system file, or other system object is described on a separate page. The reference pages are divided into seven sections, as shown in Table i. When referring to reference pages, this document follows a standard UNIX convention: the name of the command is followed by its section number in parentheses. For example, `cc(1)` refers to the `cc` reference page in Section 1.

Table i shows the reference page sections and the types of reference pages that they contain.

Table i Outline of Reference Page Organization

Type of Reference Page	Section Number
General Commands	(1)
System Calls and Error Numbers	(2)
Library Subroutines	(3)
File Formats	(4)
Miscellaneous	(5)
Demos and Games	(6)
Special Files	(7)

Release Notes

Provide specific information about the current release. Exceptions to the administration guides are found in this document. Release Notes are available online through the *relnotes* command. Each optional product or application has its own set of release notes. The *grelnotes* command provides a graphical interface to the release notes of all products installed on your system.

The IRIX Help System

Your IRIX system comes with a help system. This system provides help cards for commonly-asked questions about basic system setup and usage. The command to initiate a help session is *desktophelp*.

The Silicon Graphics World Wide Web Site

The Silicon Graphics World Wide Web (WWW) presence has been established to provide current information of interest to Silicon Graphics customers. The following URLs are accessible to most commercially available Web browsers on the Internet:

<http://www.sgi.com>

The Silicon Graphics general Web site, Silicon Surf

<http://www.mips.com>

The Silicon Graphics MIPS division site

<http://www.studio.sgi.com>

The Silicon Studio site

<http://www.ids.sgi.com>

The InterActive Digital Solutions site

<http://www.alias.com>

The Alias site

<http://www.sgi.com/Technology/TechPubs>

The Silicon Graphics Technical Publications Library

From these sites you can find all the Silicon Graphics Web-published information, including the Technical Publications Library.

Conventions Used in This Guide

These type conventions and symbols are used in this guide:

Bold User account names.

Italics Filenames, glossary entries (online, these show up as underlined), IRIX commands, manual/book titles, new terms, onscreen button names, program variables, variable command-line arguments, and variables to be supplied by the user in examples, code, and syntax statements

Fixed-width type

Onscreen text and MIP specifications

Bold fixed-width type

User input, including keyboard keys (printing and nonprinting); literals supplied by the user in examples, code, and syntax statements (*see also* <>)

""

(Double quotation marks) Onscreen menu items and references in text to document section titles

()

(Parentheses) Following IRIX commands—surround reference page (man page) section number

<>

(Angle brackets) Surrounding nonprinting keyboard keys, for example, <Esc>, <Ctrl-D>

This guide uses the standard UNIX convention for citing reference pages in the IRIX documentation. The page name is followed by the section number in parentheses. For example, *rcp*(1C) refers to the *rcp* online reference page.

User Interface Terminology

Figure i and Figure ii show examples of windows, with the window terms used in this guide noted.

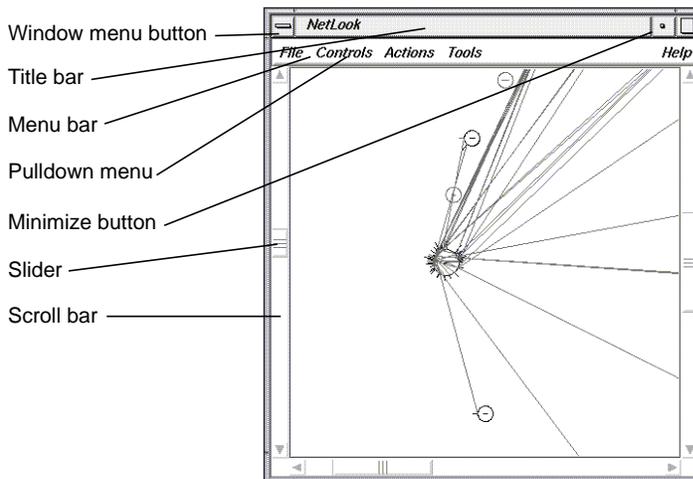


Figure i Window Terms

The mouse buttons have these functions:

- left Perform most basic tasks: click buttons, select an entry field to type into, select menu choices, select items in a display, select text to modify, and so on.
- middle Reposition windows and icons.
- right Access popup menus. Popup menus appear when you press the right mouse button in certain locations on the screen.

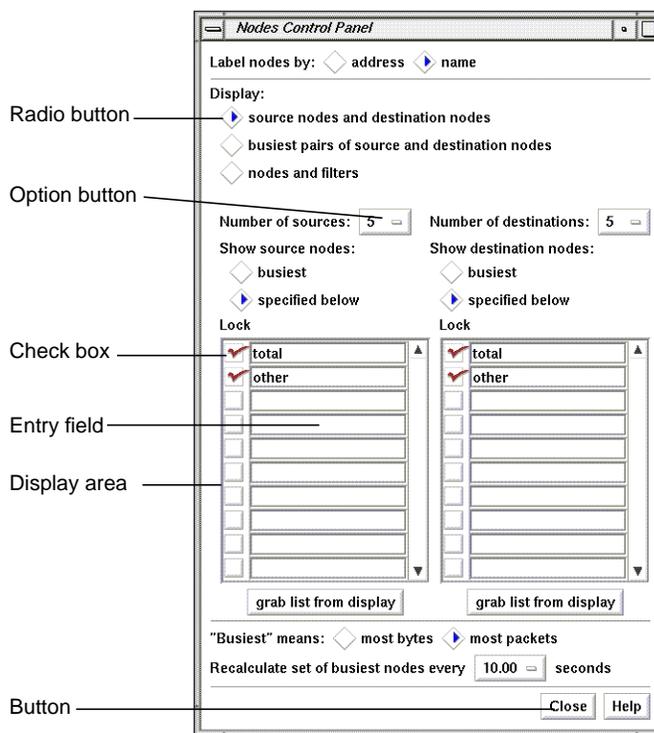


Figure ii More Window Terms

This guide uses the following terms to describe the use of the mouse:

- press Hold down a mouse button.
- drag Move the mouse while a mouse button is pressed.

click	Press a mouse button and immediately release it without moving the mouse.
double-click	Press and release a button twice in quick succession without moving the mouse.
select	The term “select” is used in the following ways: <ul style="list-style-type: none">• Click the left mouse button on an item to highlight it.• Press the left mouse button in an entry field, drag the cursor across some or all of the text, and release the mouse button. The text becomes highlighted.• Press the left mouse button on a menu title in a menu bar, move the cursor to a menu choice, and release the mouse button while a menu choice is highlighted.
deselect	Click a highlighted item to turn off the highlighting.

Common User Interface Operations

This guide assumes that you are familiar with using the mouse, working with windows, and using pulldown and rollover menus. These operations are described in the *IRIS Essentials* guide.

The sections below explain how to use additional components of the user interface that are common to several of the tools.

Using Scroll Bars

You can use scroll bars (see Figure iii) to change the area and scale of a viewing area and to display different lines or portions of lines in a display area. The size of the slider is proportional to the amount of the total that you are viewing. You operate scroll bars by pressing the left or middle mouse button when the cursor is in the scroll bar. There are several ways to operate the scroll bar:

- Press the left mouse button on the slider, drag the cursor to a new slider position, and release the button.
- Move the slider incrementally by clicking the triangles at each end of the scroll bar.

- Move the slider up or down by positioning the cursor in the trough above or below the slider and clicking the left mouse button.
- Move the slider to a specific position by positioning the cursor at that position and clicking the middle mouse button.



Figure iii A Horizontal Scroll Bar

Entering and Removing Text in a Field

Editing text in the entry fields (see Figure iv) is the same as editing text in the entry fields of other applications:

- Position the text insertion point by moving the mouse to the entry field and clicking the left mouse button.
- Select (highlight) text by pressing the left mouse button at one end of the text that you want to select and dragging to the other end.
- Select a word including a space or punctuation-delimited characters by moving the cursor to the word and double-clicking the left mouse button.
- Select the entire contents of an entry field by moving the cursor over the entry field and triple-clicking the left mouse button.
- Delete selected (highlighted) text by pressing the **<Backspace>** key.
- Delete the character to the left of the insertion point by pressing the **<Backspace>** key.

Filter:

Figure iv An Entry Field

Using Option Buttons

Option buttons (on the left in Figure v) let you select a numeric value from among a predefined set of choices. To use an option button, first press the option button with the left mouse button. A menu pops up (on the right in Figure v). Move the cursor to your selection and release the mouse button.

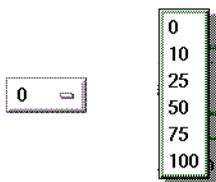


Figure v An Option Button and an Option Button Menu

Using a File Prompter

File prompter windows (like the one in Figure vi) are used to specify filenames. You can choose a filename by double-clicking a name in the display area. You can also type the name into the filename entry field and press **<Enter>** or click the *Accept* button to complete your filename selection. You can change directories to the parent of the current directory by clicking the *Up* button, or return to the directory where you started the tool by clicking the *Original* button.



Figure vi A File Prompter Window

Using Online Help

The *propel* application provides many online help files to help you as you learn to use the tools. You access these files from the Help menu in the menu bar of many *propel* windows (shown in Figure vii on top) and from the *Help* button that appears in some *propel* windows (on the bottom in Figure vii).

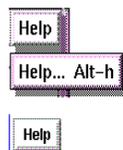


Figure vii A Help Menu and a Help Button

When you choose “Help...” from a menu or click a *Help* button, a Showcase window appears and displays the first help card.

Some help files contain several cards. Page through these cards using the <Page Up> and <Page Down> keys in the cluster of six keys just to the right of the <Backspace> key or

click the left mouse button on the arrows at the bottom of the pages. Make sure the cursor is in the Help window when you press these keys.

When you're finished reading a help file, you can close the Help window just as you close any other window, for instance, by double-clicking the Window menu button in the upper left corner of the window or by selecting Quit from the Window menu.

Propel Administration

The *propel* distributed system management application provides a convenient graphical interface to several system information databases. Support is also included for using these databases to disseminate software and system configuration file changes across your network. Using *propel*, you can automatically keep host systems updated to the correct revision level of applications or system software, and manage your system configuration files with ease.

This chapter describes the *propel* software package. The following topics are covered:

- Installation tips for *propel*. See “*propel* Installation Instructions” on page 2.
- Installation instructions for SNMP. See “Running *snmpd* on Your Network” on page 10.
- The *propel* system overview. See “The *propel* Management System” on page 11.
- An overview of the way *propel* operates. See “How *propel* Works” on page 13.
- The *propel* databases, which hold information about the systems that receive files and the files themselves, as well as the rules that govern the distribution. See “Using the *propel* Databases” on page 17.
- The Host database editor. See “The Host Database” on page 17.
- The Collection manager. See “The Collection Database” on page 22.
- The File database editor. See “The File Database” on page 29.
- The Distribution rule editor. See “The Distribution Rule Database” on page 33.
- The User database editor. See “The User Database” on page 36.
- The User Group database editor. See “The Group Database” on page 40.
- Using advanced techniques to query the *propel* databases. See “Additional Database Query Techniques” on page 47.
- Using the *propel* databases from a command line program. See “Using the *dbredit* Utility” on page 52.

- Information necessary to run the *propel* program automatically or from the command line. See “Running propel” on page 54.
- The *propel* file generation system to dynamically generate files for distribution. See “The propel File Generation System” on page 55.

The purpose of *propel* is to allow the site administrator to maintain a master set of system configuration and software distribution files on a single server (or graphics workstation), and to distribute those files to other systems throughout a heterogeneous network. The *propel* package consists of two main parts:

- A set of databases that hold information about users, network host systems, files that are to be distributed, and the set of rules that governs the distribution.
- A set of commands to maintain and modify the *propel* databases and execute the distribution.

propel Installation Instructions

Use the standard Silicon Graphics installation utility *inst(1M)* to install the *propel* software. Your system must be running IRIX version 6.2 or later for the *propel* installation to succeed.

The *propel* application includes a database and programs to edit the database. Plus, a number of sample scripts are included to load the databases by parsing system files. Individual sites will want to change these scripts to fit their configuration.

The propel Databases

The *propel* application maintains various databases to hold and query the information specific to the application’s needs.

The Host Database

The Host database holds the following information about each host on your network:

Hostname	This field holds the name that uniquely identifies a system within an administrative domain. For example, the name of the system might be <i>Wilmer</i> . This should match the system hostname in <i>/etc/sys_id</i> . The system named in this field is called the <i>named system</i> in all descriptions of other fields in the <i>propel</i> databases.
Domain	This field contains the NIS/BIND domain name of the named system.
Aliases	This field contains a list of alternative names for the named system. These names must each also be unique within the administrative domain. For example, an alias for the named system might be <i>labfileserv</i> .
IP Addresses	This field holds a list of software IP addresses for the named system. For example, software IP addresses have the form <i>123.45.67.890</i> . If the named system has more than one network connection (as in the case of a network gateway or hub), then this field should contain all the addresses used by the system.
Lease	This field is not used by the editor, but exists for the use of other applications. The field holds information about the possible expiration time of the system's IP address.
Location	This field holds the location of the named system. For example, the location of a system might be "Upstairs Lab."
Mail Exchangers	This field holds the list of system names that can receive mail and hold it for the named system. For example, the list of mail exchanger systems might be: <i>stuart.eng, gordon.eng</i>
MAC Addresses	This field holds the network hardware addresses of the host. For example, this field might contain information similar to <i>00:00:A7:10:91:03</i> . As with IP addresses, there is a different network hardware address for each interface on the system.
Manufacturer	This field describes the named system by the name of the hardware manufacturer. For example: <i>SGI</i> . The purpose of this field is to allow you to query the database and locate all systems by a given manufacturer for software update or inventory.

Model	This field contains the manufacturer's model designation of the named system. For example: <i>Indy</i> . The purpose for this field is to allow you to query the database and locate all systems of a given model.
OS Release	This field specifies the operating system software revision level on the named system. For example: <i>IRIX 5.1</i> . The purpose of this field is to allow you to locate all systems (perhaps of a given manufacturer and model) at the same operating system software revision level. You do not necessarily have to enter the operating system revision level in this field; you could also use it to track the revision level of any software you choose.
Owner	This field names the person who is responsible for the named system. For workstations, this person is usually the primary user, and for servers the usual person is the administrator who acts as a point of contact for the server users (for example, <i>Wilmer McLean</i>).
Server	This field identifies the slave distribution server system that distributes new software to the named system with the <i>propel</i> software distribution facilities. (For example, the slave server might be called <i>distmachine</i> .)
System ID	This field contains the serial number of the system hardware.
Comment	This is an open field that can be used for any purpose. It can be useful in queries to further categorize systems that are administered in a similar manner, such as development or test systems.

The standard system host databases, such as */etc/hosts*, */etc/ethers*, and the *named* maps in */var/named* can all be generated using the Host database as input. Under *propel*, administrators need only maintain one list of host information for the entire network. Individual system data files can be generated from this list, thus simplifying large domain host administration. See "The propel File Generation System" on page 55 for complete instructions.

The Collection Database

The Collection database holds records defining collections of hosts from your Host database. These can be arbitrary groupings, or groups generated by querying the Host database. The following fields are used in the Collection database:

name	The name of the collection of hosts. Make this name unique from any hostname. For example, if you wish to make a collection describing all Silicon Graphics systems on your network, you might select the name <i>sglhosts</i> .
------	--

collection	The query parameters to define the collection. You can query on any field or fields in the Host database. For a complete set of instructions on forming a database query, see the section titled “Additional Database Query Techniques.”
comment	Any notes you wish to make about the collection. This is an open field that can be used for any purpose.

Information on creating, modifying, and querying entries in the Host and Collection databases is found in “The Host Database” on page 17 and in “The Collection Database” on page 22.

Initializing the Host Database

When you install *propel*, the host database is initialized automatically. If you wish to reinitialize the database at any time, enter the following command while logged in as **root**:

```
dbredit host init
```

Remember that this command will eliminate all current entries in your Host database.

Automatically Loading the Host Database

All systems using the *propel* software must have a record in the *propel* Host database. The easiest way to place all your systems in the database is to parse the standard system files */etc/hosts* and */etc/ethers*, and load the system names directly into the database. There are a pair of scripts provided with the *propel* software that parse these files and load the database. The first is a *perl* script, */usr/propel/sample/host_parser* that reads through these system files, and outputs one line for each system name that it finds. The output format is that of a TCL keyed list, which can be inserted into the host database using the *dbredit* command. Running the script */usr/propel/sample/add_hosts* works for many sites. You may edit the *host_parser* script to make changes to fit your local needs.

The *propel* database has one entry for each “host,” but each host can have multiple network interfaces, and therefore multiple entries in the *hosts* file. The standard *host_parser* script recognizes the extra interfaces of hosts if they have a comment of the format *GATE(hostname)* or if the primary name for these interfaces is of the form *gate3-hostname*. The test for such names is clearly marked within the script and can be changed to fit local conventions.

The most useful script available for populating your *propel* Host database is */usr/propel/sample/find_hosts*. Invoked with an IP subnet number (in the form 123.45.67) as an argument, this script sends an ICMP echo (*ping*) request to each possible node (0-255) on the specified subnet. When a remote host responds, it sends an SNMP request to the remote host and creates a record in the Host database with any data that is returned.

Once a record exists in the Host database for each host, it is a good idea to fill in the remaining information for each host. This can be done by completing each entry in an editor, or as a batch update. There is a very simple script that attempts to attach to each system using the *rsh*(1) command, and reading information from the *uname*(1) command to glean the standard information. This script is */usr/propel/sample/host_info*, and it outputs a line for each system that successfully completes the process. There is a simple *perl* wrapper program, */usr/propel/sample/batch_update*, that takes a file of *host_info* output lines and performs the updating in the Host database. A more powerful script, */usr/propel/sample/update_hosts*, updates the records of each host in the database by parsing the result of an SNMP request for the *sysDescr* variable. If you are running SNMP on the client hosts then this script should provide the best automatic updating.

If you are administering a network of Silicon Graphics systems, the following steps will initialize and populate your Host database. If you are administering a heterogeneous network, not all hosts may be located and entered in the database through this process. If this happens, you may need to edit the databases with the *hostedit* editor, as described in "Editing the Host Database." Follow these steps to populate your database:

1. Certify that the entire distribution has been installed on your master server system and log in as **root** or as a member of the user group **irixpro**.
2. Obtain a basic listing of the hosts on your network from the */etc/hosts* and */etc/ethers* files with the command:

```
/usr/propel/sample/add_hosts
```

3. Next, use the command

```
hostedit
```

to view and edit the Host database entries for completeness and correctness. You may wish to create a collection (see "The Collection Database" on page 22 for more information about creating collections) of all your Silicon Graphics systems at this time if you need to distribute the SNMP agent software to them.

4. Certify that an SNMP agent (*snmpd*) is running on each host on the network. On each host, enter the command:

```
ps -ef | grep snmpd
```

5. If Silicon Graphics systems need to have *snmpd* installed, use the */usr/propel/sample/snmp_push* script. This script adds File database records for all files needed to run *snmpd* and *hpsnmp* on a host.

After running the *snmp_push* script, you can enter the command

```
runpropel snmp-6.2 `os="IRIX 6.2"``
```

to push SNMP to all Silicon Graphics systems running IRIX 5.3 or later. You can use any host query to select the correct hosts in your domain.

For non-Silicon Graphics systems, you must make sure that a compatible SNMP daemon is running. Consult your documentation for each system to determine how to obtain an SNMP daemon.

6. Finally, use the command

```
update_hosts
```

to use SNMP to obtain any remaining fields from your SNMP hosts.

You can also use the */usr/propel/sample/find_hosts* script, which performs many of the same functions. This script searches your network for other hosts and populates the database using SNMP responses. Obviously this command is only useful when the SNMP daemon is running on all your systems. The command has the syntax:

```
find_hosts subnet
```

The script pings each possible address on the given subnet. For example, if given the subnet argument 127.0.0, *find_hosts* returns the following:

```
Trying: 127.0.0.1
  Address 127.0.0.1 pings, but is not in database.
  Does not answer SNMP request.
Trying: 127.0.0.2
Trying: 127.0.0.3
Trying: 127.0.0.4
```

And so on up to address 127.0.0.255. IP address 127.0.0.1 is the standard loopback interface. The *ping* response received was from the local host. When this command is used on a valid subnetwork, the database is populated using the SNMP responses from the hosts. You must be logged in as **root** to use *find_hosts*.

Creating the Collection Database

Once you have populated your Host database, you should create your first set of collections of hosts. Instructions for creating and modifying collections are found in the section titled “The Collection Database” on page 22.

Populating the User and Group Databases

The script used to automatically populate both the User and Group databases at once is */usr/propel/sample/add_users*. This script parses the */etc/passwd* and */etc/group* files to build these databases. The script is sophisticated enough to bypass NIS entries in these files, but does generate entries for accounts such as *uucp* and demonstration users.

As **root**, enter the command:

```
/usr/propel/sample/add_users
```

Loading the File Database

Initially, you must create a set of files that can be distributed to client hosts. The creation process includes putting the files into the */var/propel/data* directory, and creating a File database entry for each file. Files can also be automatically generated before they are distributed by setting a generator command in the File database record. Scripts are included in your propel application for generating the files */etc/hosts*, */etc/ethers*, and the DNS maps, using information from the Host database. Each of these sample scripts must be edited to deal with local naming conventions at your site.

Files can be distributed from locations other than the */var/propel/data* directory if slave servers are not used, that is, only if files are distributed from the original server system directly to the final client systems. By default, only the *data* directory can be distributed to slave servers, which subsequently distribute the files to client systems. This behavior can be changed, however, by editing the */usr/propel/bin/runpropel* script.

The propel License

The licensing for *propel* is managed by *flexlm*. This license is usually specified in the file */var/flexlm/license.dat* or by a file specified by the environment variable `LM_LICENSE_FILE`.

propel and rdist

The *runpropel* program uses the industry standard utility *rdist* to distribute files from master servers to slave servers, and from slave servers to client hosts. The *rdist* utility originates from the BSD releases and is a standard utility under most UNIX operating systems. It is also freely available on the Internet in source form so that it can be ported to most other systems. The purpose in using a standard file distribution program is to make *propel* useful in a heterogeneous environment. The *rdist* utility is distributed with *propel* in a *gifts* installation package. When installed, the source is placed in the */usr/people/ADgifts/rdist-6.1.0* directory.

The *rdist* program pushes files using a generalized configuration file to control the distribution. The *runpropel* program generates the necessary *rdist* configuration files (known as *Distfiles*), and then runs *rdist* on the *Distfiles*.

All *propel* servers and clients must have the 6.1 version of the *rdistd* daemon installed, at a minimum. The *propel* software works with any version of *rdist*, but the version shipped with *propel* is *rdist* 6.1. The use of *rdist* 6.1 is strongly recommended for its extended logging, improved security, and improved performance. There are, however, some sites that cannot build the new *rdist* software but have the BSD software running. These sites must use the old distribution software on the master, and a modified master *runpropel* script. If you would like to use *propel* with an older version of *rdist*, you must first unset the LOGERRS variable in the */usr/propel/bin/runpropel* command script. To set the LOGERRS variable, edit the *runpropel* command script and change the line that reads:

```
LOGERRS="-l
syslog=warning,nerror,ferror:file=/usr/propel/rdist/log.$FREQUENCY=all:
stdout=warning,nerror,ferror -L syslog=all"
```

to read as follows:

```
LOGERRS=" "
```

Server/Client Security

The *propel* application currently uses *rdist* for its distribution mechanism, which means that every client system must trust at least one other system on the network as its server. If files are to be pushed directly from the master server to client systems, then each client system must place the master server's hostname in its *.rhosts* file. If a hierarchical push strategy is used then each client must have the hostname of the slave server from which the client will receive files in the client's *.rhosts* file, and the slave server must have the hostname of the master server in its *.rhosts* file. Files may be pushed with the UID of a user other than **root**, in which case the *.rhosts* file in that user's home directory needs to include the server's system name.

Running snmpd on Your Network

In order to use many of the utilities and features of *propel*, each system on your network should be running the *snmpd* daemon. This daemon provides support for SNMP (Simple Network Management Protocol) and allows other systems to query the host for information about its configuration.

To obtain SNMP support, the provision distribution packages to install on each station are:

```
snmpd          01/04/95 SNMP Daemon with HP MIB Support
snmpd.sw       01/04/95 SNMP Daemon with HP MIB Support
snmpd.sw.agent 01/04/95 SNMP Daemon with HP MIB Support
```

To configure a workstation so that *snmpd* is started automatically when the system is rebooted, install the packages listed above, and enter this command on the system while logged in as *root*:

```
chkconfig -f snmpd on
```

To check to see if the daemon is already running, enter this command:

```
ps -e | grep snmpd
```

If there is no output from this command, *snmpd* is not running. Become *root* and enter this command to start *snmpd*:

```
snmpd &
```

How to Use propel

The following sections explain how to get the most out of *propel* for various tasks. This is not an exhaustive list. The configurability and extensibility of *propel* allow you to modify and extend the software to meet the individual needs of your installation.

The propel Management System

There are six databases that hold the information needed by *propel*. The databases hold information about:

- Collections of related hosts
- Information about the files and file packages being sent and received
- Groups of users
- Hosts that send and receive files
- Information about the rules under which files and packages of files are distributed to hosts and collections of hosts.

- User accounts

A graphical tool and command-line interface is provided for editing each database, as well as an example script that shows how to build the initial Host database from the standard system files. The command

propel

brings up a standard desktop view of the *propel* tools. The desktop view appears similar to this, shown in Figure 1-1:

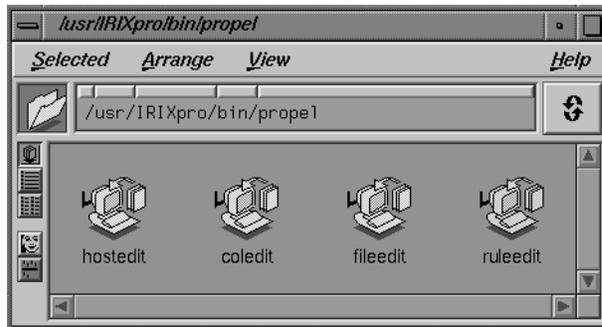


Figure 1-1 The propel Directory View

Alternately, you can invoke each tool on its own with one of the following commands:

```
/usr/propel/bin/coedit  
/usr/propel/bin/fileedit  
/usr/propel/bin/groupedit  
/usr/propel/bin/hostedit  
/usr/propel/bin/ruleedit  
/usr/propel/bin/useredit
```

These commands bring up the graphical tools for the Collection, File, Group, Host, Rule, and User databases, respectively.

Icons in any *propel* tool window can be dragged and dropped on other *propel* tool windows to fill needed fields. For example, you may populate a collection by dragging host icons from the Host database editor and dropping them in the appropriate window of the Collection manager.

The command line interface for editing databases is *dbredit(1M)*. This command allows you to edit the database of your choice. The *dbredit* utility is completely described in the

dbredit(1M) reference page, but an overview is also provided in this guide in the section titled “Using the *dbredit* Utility” on page 52.

How propel Works

The *propel* software package is highly flexible. In fact, the individual administrator who uses *propel* defines virtually every parameter of the operation. As an example, *propel* does not come with a preconfigured set of rules concerning the frequency with which files are propagated. The terms “Daily” and “Weekly” mean nothing to *propel* until you define them. You enter terms as parameters in the *propel* databases, and then use different commands to query the databases and execute the distributions based on those terms. It is therefore vitally important that you plan your *propel* strategy in advance to avoid confusion.

It is this flexibility that makes *propel* a powerful tool. Because there is not a predefined set of rules about concerns such as the frequency of the distribution and the naming conventions of files and packages of files, you are not restricted in your use of *propel* to simple hourly, daily, or weekly updates, though these might form the bulk of your usage.

You will probably start your *propel* databases by listing the host systems to which you will distribute files. This is a straightforward process of listing the names of the systems and their network addresses and other pertinent information about them. This is where you start defining your *propel* environment. The fields in the Host database that describe the manufacturer, Operating System revision level, and other such information can be vitally important when you wish to target a very small subset of a great number of hosts for distribution. There are no preset entries for these fields, so you can use them (especially the comment field) as you wish. The *Fill* button on the *hostedit* window is provided to assist you in populating your database. If you list the hostname or IP address of a workstation on the network and use the mouse to click the *Fill* button, *propel* uses SNMP to query the host itself for answers to as many of the other fields as possible.

The Collection database is used to make collections of hosts from the Host database. This is done through making queries to the Host database. All hosts that match the query parameters you select will compose the collection. To prepare for these queries, it is recommended that you fill in all fields in each Host database record.

The User database allows you to list all the users at your site and manage that information from a central server. Since all user information is managed at one place, you can be sure that user names and UID numbers are not duplicated, and that departing users will be removed from all systems at the site. You can also use this database to provide access to new systems for existing users. The */etc/passwd* and */etc/group* files can be generated for and distributed to any system on your network by this database.

The Group database performs the same function for users that the Collection database provides for hosts. This database allows you to make groups of users with related needs or other common factors. The *groupedit* tool allows you to edit and visualize your user groups.

The File database allows you to list the files that are to be distributed in much the same manner that you listed your host systems and users. There is a field in each file entry called "Package" that lists the name of a file package (a group of related files.) There can be more than one entry in the File database for each file you wish to distribute. You may wish to place a file in more than one package, or you may wish to have a file generated by one program for some hosts and another program for other hosts. In these cases, you make two entries in the File database that both point to the same file, but with different file-generating programs or in different packages.

The most flexible of all the tools is the Rule database. By itself, the Rule database takes no action. The *runpropel* command searches the Rule database for the parameters given on the *runpropel* command line and executes any rules that match those command line parameters. Thus, you may create a rule that says that files A and B and file package C are to be distributed to host collection "enr" and to any client host with a hostname beginning with the letter "Q," and the frequency of the distribution is to be "walnuts." That frequency has no meaning until you assign it a meaning by the way you use *runpropel*.

Typically, distributions of a regular frequency are managed through the *cron(1)* facility of your workstation or server. The *cron* utility does define regular time frequencies, so it is possible to simply tell *cron* to execute the command

```
runpropel -f walnuts
```

every day at exactly 2 PM. Thus, "walnuts" means "daily at 2 PM" because you set it up to have that meaning. This may seem like a lot of work to perform a simple task, but the advantage comes when you want to change the meaning of "walnuts." You then have only to change your *cron* command, and not the database entries of every rule that runs with a frequency of "walnuts."

Once you become comfortable with the concepts behind *propel*, you can configure the software to suit your needs exactly. The graphical tools make this configuration easy and intuitive. There is a command line interface available through the *dbredit(1M)* command for server users with no graphics capability.

Using *propel* to Manage a Domain

This section contains some information on using *propel* to simplify the management of a large network domain.

Distributing Silicon Graphics Software With *propel*

Silicon Graphics currently releases all software in a proprietary format. This format can only be installed using the *inst(1)* program. In the past it has been very difficult to upgrade a large number of hosts with *inst* because it required that the host be running from the *miniroot*, and required a great deal of interaction with the administrator. New versions of *inst* support live installations, and installations from a configuration file of preselected options. This allows the administrator to “walk through” an installation on one host, saving his or her choices into a configuration file, and then performing identical installations on other hosts while the hosts are running, with no interaction. If the configuration file is checked into the *propel* File database and distributed to a group of systems with the *inst* command specified as the exit operation, an entire network can easily be upgraded from a central host with *propel* invoking *inst* on several hosts at the same time, and logging errors for any installation that fails. To perform a mass installation, distribute the *inst* configuration file, and specify a command such as:

```
inst -a -f system:/location_of_package -F config-file
```

For information on creating an *inst* configuration file, see the *Software Installation Administrator's Guide*.

Internal Data Distribution With *propel*

For sites that develop and use in-house programs or data, *propel* is an obvious choice for keeping all the hosts on a network at the same software revision level. Since *rdist* is supported on nearly every version of UNIX in existence, *propel* can instantly be installed and used to manage large networks of heterogeneous systems.

Installing Free Software Packages With propel

Typically, packages obtained from public networks have the ability to install themselves into the non-standard directory of your choice. For instance, all of the tools from the Free Software Foundation have a `DISTDIR` variable in the *Makefile*, which specifies the top level directory in which the software is to be installed. For example, if you install the GNU C compiler into the directory `/var/propel/data/gcc`, you can add a file database record for the directory, then distribute the directory to client systems. For example, you could install the `gcc` directory onto all of your Indigo and Indy systems by giving the command:

```
runpropel gcc `model~Ind*`
```

You can also keep your client systems up to date with the `gcc` software by adding a distribution rule to update them weekly, and then by occasionally deleting the old `gcc` package, and by installing a new version of `gcc`.

Host Information Management With propel

The *propel* application uses a full-featured database to maintain information about hosts in your domain. The Host database is automatically locked and unlocked during operations so that many users can update the information concurrently, and the database deals with complicated queries quickly. Enough information can be kept in the *propel* Host database that all of the standard host files can be generated using the database, centralizing all host management into a single location. Scripts are included to create the system files `/etc/hosts`, `/etc/ethers`, and all of the DNS databases. The Host database can also be used to generate and distribute the DBM databases for the optional NIS software.

User Information Management With propel

The User database performs the same function for your network's users that the Host database performs for your hosts. Each user has an entry that can be copied to any new system to which that user needs access. Furthermore, you can load the User and Group databases, you can generate the `/etc/passwd` and `/etc/group` files, and you can create user accounts with the following scripts found in `/usr/propel/sample`.

<code>account</code>	Generates accounts for a list of users by using <i>propel</i> to push a copy of the local <code>guest</code> account to the remote host, and changing the permissions after the transfer is complete.
<code>add_users</code>	Loads the User and Group databases by parsing the local host's <code>/etc/passwd</code> and <code>/etc/group</code> files.

<i>make_group</i>	Generates a file in standard <i>/etc/group</i> format from the <i>propel</i> Group database.
<i>make_passwd</i>	Generates a file in standard <i>/etc/passwd</i> format from the <i>propel</i> User database. It takes a user query as an argument so you can include part or all of the User database in the generated file.

Localizing propel

The *propel* application is written primarily using human-readable scripts. The editors are all created with the SGITCL programming language and an SQL database. The user interface routines are all created using the Bourne Shell (*/bin/sh*), and the sample scripts are a mixture of SGITCL, *sh*, and *perl*. The only portions of the software that cannot be changed easily are the query parsers, which were built with *lex* and *yacc*, and the SGITCL interpreter, which requires a database development license.

Fields can be added easily to any of the databases, and to the editing tools by adding a small block of code. Tools can be created to maintain system files other than those already supported by existing *propel* scripts.

Using the propel Databases

The following sections explain the fields in each of the *propel* databases and offer specific instructions in the use of the various database editing utilities. A section on formulating queries is also provided. A final section offers instruction in the use of the *dbredit(1M)* command-line database editor.

The Host Database

The *Host* database contains information about all of the systems in an administrative domain. Each record in this database contains the following fields:

Hostname	This field holds the name that uniquely identifies a system within an administrative domain. This should match the system hostname in <i>/etc/sys_id</i> . The system named in this field is the named system in all subsequent fields. (For example, the name of the system might be <i>Wilmer</i> .)
----------	--

Domain	This field holds the network domain name of the system, for example, <i>lab.eng.com</i> .
Aliases	This field contains a list of alternative alias names for the named system. These names must each also be unique within the administrative domain. (For example, an alias for the named system might be <i>labmachine</i> .)
IP Addresses	This field holds a list of software IP addresses for the named system. (For example, software IP addresses have the form 123.45.67.890)
Lease	This field is not used by the editor, but exists for the use of other applications. The field holds information about the possible expiration time of the system's IP address.
Location	This field holds the location of the named system. For example, the location of a system might be "Upstairs Lab."
Mail Exchangers	This field holds the list of system names that can receive mail and hold it for this host. For example, the list of systems might be <i>stuart.eng</i> , <i>gordon.eng</i> .)
MAC Addresses	This field holds the network hardware address of the host. For example, this field might contain information similar to: 00:00:A7:10:91:03
Manufacturer	This field describes the named system by the name of the hardware manufacturer. (For example, <i>SGL</i> .)
Model	This field contains the manufacturer's model designation of the named system. (For example, <i>Indigo</i> .)
OS Release	This field specifies the operating system software revision level on the named system. (For example, <i>IRIX 5.1</i> .)
Owner	This field names the person who should be held responsible for the named system. For workstations, this person is usually the primary user, and for servers the usual person is the administrator who acts as a point of contact for the server users. (For example, <i>Wilmer McLean</i> .)
Server	This field identifies the slave distribution server system that should update the named system. (For example, the slave server might be called <i>distmachine.eng</i> .)
System ID	This field contains the serial number of the system hardware.

Comment This is an open field that can be used for any purpose.

The standard system host databases, such as */etc/hosts*, */etc/ethers*, and the *named* maps in */var/named* can all be generated using this database as input. Under the *propel* distribution system, administrators need maintain only one list of host information. System data files can be generated from this list, thus simplifying large domain host administration. For more information on file generation using *propel*, see “The propel File Generation System” on page 55.

Editing the Host Database

To edit the Host database, enter the command:

```
/usr/propel/bin/hostedit
```

You see the following initial screen, shown in Figure 1-2:

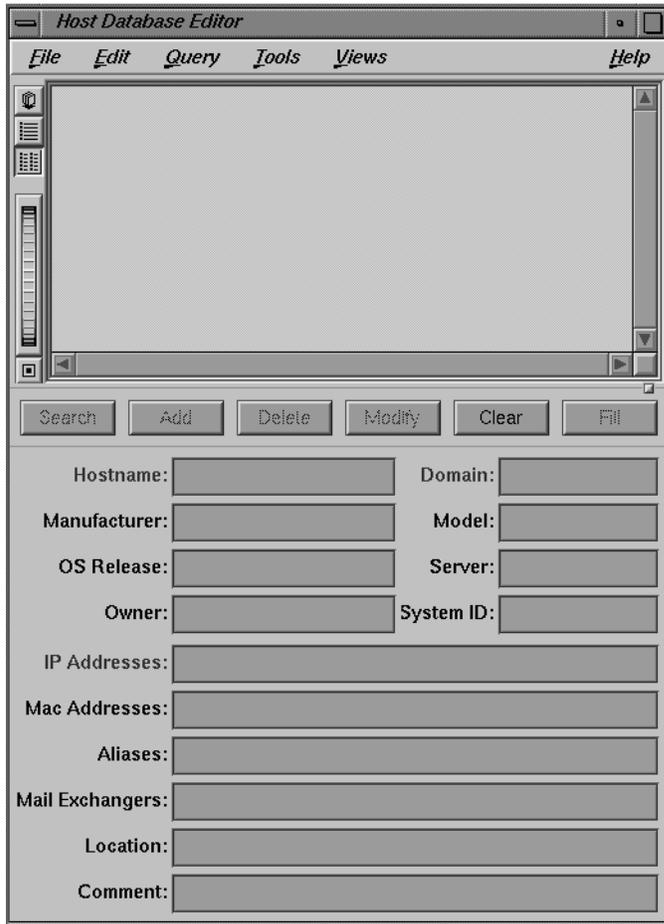


Figure 1-2 The Host Database Editor

When you first see this screen, there are no hosts listed and only the *Clear* button is functional.

To add a host to your database, you must fill in the following fields (which appear in red in the *hostedit* window):

- Hostname
- IP address (at least one)
- Domain

When there are entries in these three fields, the *Add*, *Search*, and *Fill* buttons become functional. You can see a change in their visual presentation. If possible, you should enter all the fields. Once entered, you can use each field as a query parameter for that record in the Host database.

Inserting a Host Entry

Once you have entered the host's information, press the *Add* button by placing the cursor over the button and pressing the left mouse button. You see the information appear in the window in a slightly different format. This tells you your entry has been accepted into the database. If you do not know all the information, you can also use the *Fill* button to automatically query the system itself for various fields, such as the MAC address and OS version fields, before you add the entry. The *Fill* button works only if the remote system is running the *snmpd* daemon. See "Running *snmpd* on Your Network" for information on starting *snmpd* on a remote system.

Updating a Host Entry

To change this entry, click the entry in the window with the left mouse button to highlight it. The information you entered appears in the fields in the lower part of the screen. Change these entries as you wish and then press the *Modify* button.

Querying the Host Database

For an example of using the Host database fields as query points, you can make a query to the database to find all systems that are owned by user *Kate* and are at OS Release level 5.3. Begin by using the mouse to press the *Clear* button. This clears the fields of all information. Then enter your query parameters in the appropriate fields. In this case, enter "*Kate*" in the *Owner* field and 5.3 in the *OS Release* field. You can use glob expressions as wild cards to search in any field. For more information on using glob expressions in queries, see the section titled "Additional Database Query Techniques." When you have made your query, use the mouse to press the *Search* button. All hosts in your database that match the specified parameters will be displayed in the window. If

you know that there is only one system owned by any individual, you can use the *Fill* button to display the remaining fields.

Parsing System Files Into the Host Database

A tool is provided to allow you to quickly parse your standard system files into the Host database. This tool is *add_hosts(1)* and is found in the directory */usr/propel/sample*. The *add_hosts* command parses your system's existing */etc/hosts* file and inserts a basic record for each host found into your *propel* Host database. The *update_hosts* utility uses *snmpd* to fill in the remaining fields for each host running *snmpd*. For step-by-step instructions on filling your Host database using these programs, see "Automatically Loading the Host Database" on page 5.

For more information on running *snmpd* on all your hosts, see the section titled "Running snmpd on Your Network."

The Collection Database

Records in the Collection database have the following fields:

- | | |
|------------|--|
| name | The name of the collection of hosts. Make this name unique from any hostname. |
| collection | The query parameters to define the collection. You can query on any field in the Host database. The field is made up of a list of typed fields separated by newlines (<Enter> keystrokes). The first word on each line is the type of query that the line represents. The available types are <i>hostname</i> , <i>hostquery</i> , and <i>collection</i> . |
| comment | Any notes you wish to make about the collection. |

Editing the Collection Database

The *coedit* utility allows you to define collections of hosts for *propel*. A collection of hosts is simply a group of hosts that you have defined. Usually, collections are made of hosts with some common factor, such as manufacturer or usage. Since you specify the attributes of a host by using the *hostedit* utility, you can prepare hosts to be part of a collection when you enter them in your Host database. For example, you can enter several hosts in your Host database with the comment "engineering lab" in the comment field. Then, when you use *coedit* to create a collection, you specify the "engineering lab" comment as the query field when creating the collection. The syntax for these queries is

described in “Additional Database Query Techniques” on page 47. All hosts that you entered with that comment becomes part of the collection. Then you name the collection, and use the Rule database to specify how and when the collection should receive files. The names of collections and the Host database queries that define the collection are stored in the Collection database.

Once defined, the *coledit* utility stores the parameters of the collection in a database. Enter the command:

```
/usr/propel/bin/coledit
```

and you see the following window, shown in Figure 1-3:



Figure 1-3 The Collection Manager

When you bring up the Collection Manager window after your database is populated, you see an icon and a name for each collection in the database.

Adding a Collection of Hosts to the Collection Database

To use the *coledit* tool to add a host collection to your Collection database, place the mouse cursor inside the *coledit* window and use the right button of the mouse to bring up the popup menu. There are several options:

Help Brings up a help card.

Add Collection Brings up the Add Collection window.

- Add Host Adds a host to an existing collection.
- Add Query Brings up the Add Host Query window, and allows you to add an additional query parameter to select hosts for an existing collection.
- Remove Selected
 Removes the selected collection.
- Edit Selected Hosts
 Brings up the *hostedit* tool with the selected host's information displayed. To select a host, you must have a collection displayed in a collection display window.
- Preview Shows you the list of hosts currently selected in a collection.

Select the Add Collection option to create a new collection of hosts. When you select this option, you see the new window, shown in Figure 1-4.



Figure 1-4 The Add Collection Window

You must enter a name for the collection, then use the left mouse button to press the *Accept* or *Apply* button to create your new collection. The *Apply* button creates your new collection, and retains the Add Collection window, so you can continue to create more collections conveniently. The *Accept* button creates your collection and removes the Add Collection window. The *Cancel* button cancels the operation without creating the new collection.

Once you have created the collection, you must specify hosts to be part of the collection. You should notice that an icon has appeared in the *coledit* window. If you made two collections and called one "first collection," and the other "second collection," the window now looks like Figure 1-5.



Figure 1-5 A Populated Collection Manager Window

To add a list of hosts, a list of collections, a host query, or any combination of these things to your new collection, double-click the left mouse button with the mouse cursor over the icon for the new collection. You see a new window that looks something like Figure 1-6.

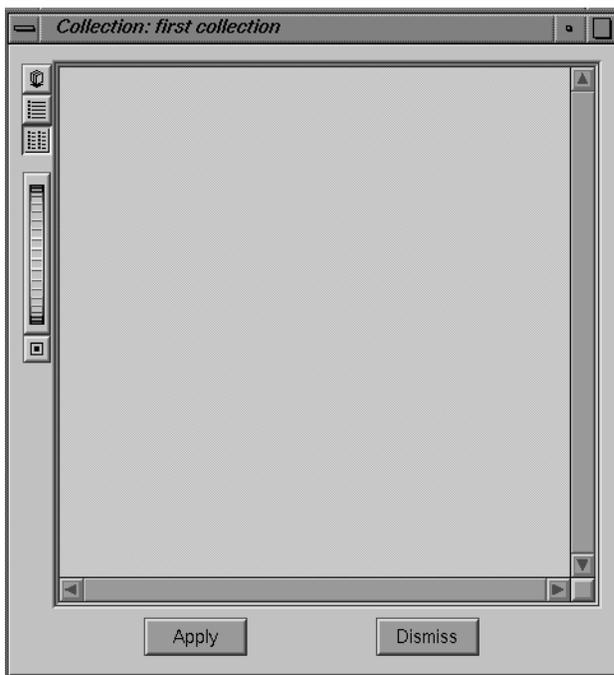


Figure 1-6 A Collection Display Window

Now use the right mouse button to bring up the popup menu, and select either the Add Host, Add Collection, or Add Query options to populate your collection.

The Add Host option brings up the Add Host window, shown in Figure 1-7, with the entire list of hosts from your Host database.



Figure 1-7 The Add Host Window

To place a host in your new collection, place the mouse cursor over the name of the host you wish to select and double-click the left mouse button. When you are finished selecting individual hosts, click the *Accept* button to secure your changes and remove the Add Hosts window.

The Add Collection option brings up the Add Collection window, previously shown in Figure 1-4. All your currently configured collections are shown in this window (although in this example, you are creating your first collection, so no collections may appear at this time). As with the Add Host window, double-click the name of a collection you wish to add to your new collection and complete your changes by pressing the *Accept* button.

The Add Query option brings up the Add Host Query window, shown in Figure 1-8.



Figure 1-8 The Add Host Query Window

This window allows you to enter a query to the Host database. You must formulate your query according to the rules outlined in “Additional Database Query Techniques” on page 47.

When you have used these various windows to populate your collection, the collection display window looks something like the window shown in Figure 1-9.

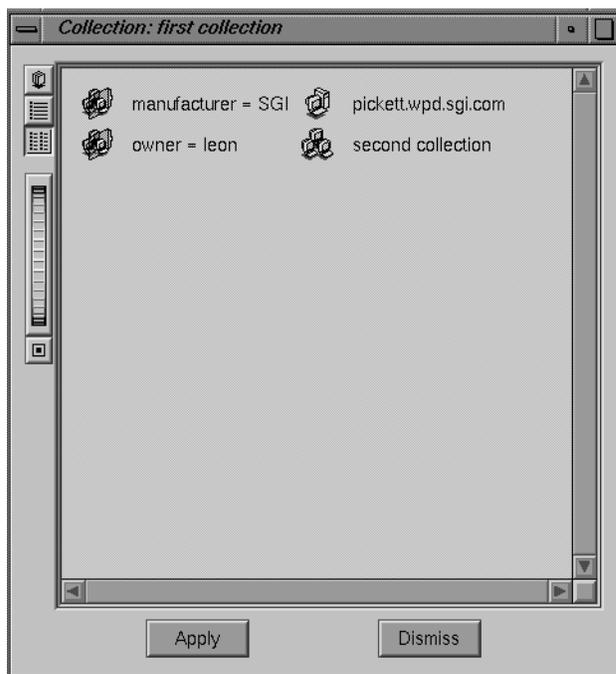


Figure 1-9 A Populated Collection

The collection shown in Figure 1-9 has a single host, a pair of host queries, and another collection as its elements. This is to demonstrate the various items that may go into a collection.

Once you have entered all the information and created your collections, press the *Apply* button in the Collection Manager window to place your collections in the Collection database.

Updating a Collection Database Entry

To change an entry, click the icon with the left mouse button to highlight it, and the Collection Display window for that collection appears. Change these entries as you would if you were making a new collection, then click the *Apply* button to finalize your changes.

The File Database

The File database contains all the necessary information about the files that are to be distributed to systems on your network.

Records in the File database have the following fields:

Source	The field usually contains the name of a file in the <i>/var/propel/data</i> directory. The file specified in this field becomes the named file in all subsequent fields in the database entry. (For example, the file named might be: <i>/var/propel/data/usr/local/bin/progname</i>). Any file on your system can be specified as a source file to be distributed directly to clients, but only files from the <i>/var/propel/data</i> directory structure can be distributed to slave distribution servers for further sub-distribution to clients. Always specify the full (absolute) pathname of each source file in this database.
Generator	If this file is generated, the name of the program that generates the file goes here.
Destination	This field specifies the pathname on the remote system where the named file will be placed (for example, this field might contain the pathname: <i>/usr/local/bin/progname</i>).
Pre-OP	This field specifies any commands to be run on the remote system before the named file is installed (for example, this field might contain the command: <i>/etc/killall progname</i>).
Post-OP	This field specifies any commands to be run on the remote system after the named file is installed (for example, this field might contain the command: <i>/usr/local/bin/progname</i>).
Exit-OP	This field contains any commands to be run on the remote system after the last file in the distribution rule is distributed. For example, if certain daemons need to be restarted to read distributed configuration files, this field should contain the command to restart them.

Package	This field holds the name of the software package that the named file belongs to, if such a package exists (for example, the named file could be part of a package called <i>pkg.1</i>).
Comment	This field is available for any purpose.

Editing the File Database

This database contains the names of the files that are propagated to systems in your *propel* network. The File database also contains instructions for commands to be run on the client system before and after the file is transmitted, and information about file packages of which this file is a member.

The command `/usr/propel/bin/fileedit` invokes this tool, and you see the following screen, shown in Figure 1-10.

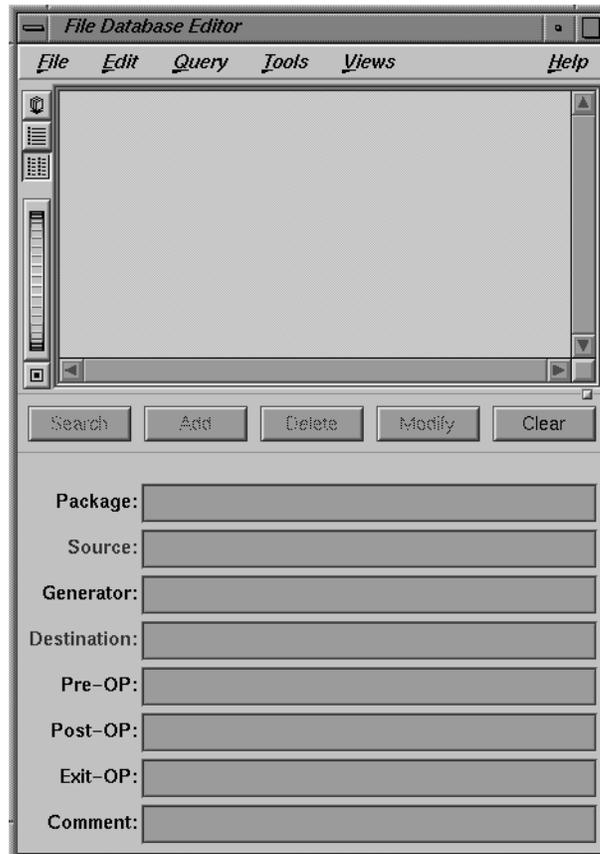


Figure 1-10 The File Database Editor

Adding a File to the File Database

To use the *fileedit* tool to add a file to your File database, you must enter the following information in the fields provided:

- Source pathname
- Destination pathname

The remaining fields are optional, but you should fill them in, if at all possible, for use in file queries. The source pathname field is where you list the location of the file to be propagated. Typically, all files are located in the directory */var/propel/data*. If you have many files and file packages to propagate, this directory may have many subdirectories or may be linked to a separate directory altogether. Only source files in this directory structure may be distributed to slave distribution servers for further distribution to client hosts. If you are distributing a file directly to the final client host, you need not have the source file in this directory structure, though it is recommended for convenience and organization.

Once you have entered the file's information, press the *Add* button by placing the cursor over the button and pressing the left mouse button. You see the information appear in the window in a slightly different format. This tells you that your entry has been accepted into the database.

To obtain a listing of all source files, packages, or destination locations in your database, use the right mouse button to bring up a popup window with the cursor over the appropriate field. For example, if you choose the List option from the popup menu over the Source field of the File database editor, you see a window very much like that shown in Figure 1-11.



Figure 1-11 A List of Source Files in the File database

Updating a File Database Entry

To change this entry, click the entry in the window with the left mouse button to highlight it, and the information you entered appears in the fields in the lower part of the screen. Change these entries as you wish and then press the *Modify* button.

Querying the File Database

For an example of using the File database fields as query points, you can make a query to the database to find all files that are part of the package "network configuration" and have the comment "fully propagated." Begin by using the mouse to press the *Clear* button. This clears the fields of all information. Then enter your query parameters in the appropriate fields. Then use the mouse to press the *Search* button. All files in your database that match those two parameters will be displayed in the window.

The Distribution Rule Database

This database holds a set of rules that control the distribution of files and packages to host systems. This file uses a very simple rule set to describe which files should be pushed to which systems, and under what conditions the push takes place.

The fields in this database are:

Frequency	This field holds a simple term of your choice to describe how often files and packages with this frequency are pushed. For example, you may set your frequencies to Daily and Weekly, and run them accordingly with <i>cron</i> , or you can simply call the frequencies A, B, and C and select your own definitions of those terms through <i>cron</i> or manual use of the <i>runpropel</i> command. The <i>runpropel</i> command line that you issue determines the actual operation; this field is merely your tag for a frequency group that you define.
Name	This field contains a name for the distribution rule.
File Query	This field contains a file attribute query. Use this field to enter keywords to find in the File database.
Host Query	This field contains a host attribute query. Use this field to enter keywords to find in the Host database.
Comment	This is an unused text string that can be used for noting information about a rule.

Editing the Rule Database

The *ruleedit* command allows you to specify which files or packages of files are pushed out to a list of client hosts. The Rule database stores the rules you have created with the *ruleedit* command. Enter the command:

ruleedit

You see the following screen shown in Figure 1-12.

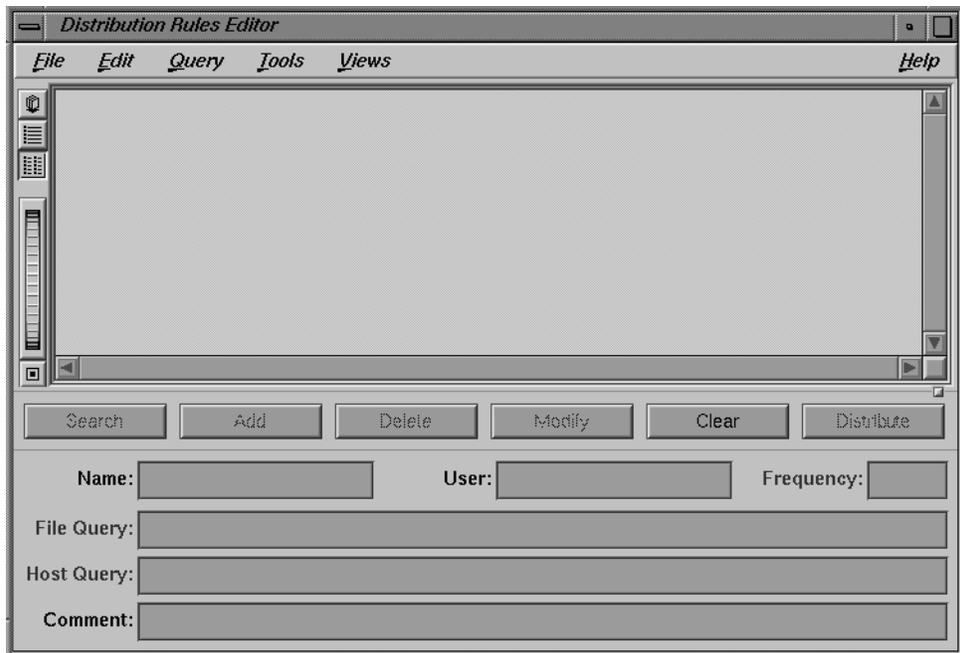


Figure 1-12 The Distribution Rules Editor

Adding a Distribution Rule to the Rule Database

To use the *ruleedit* tool to add a distribution instruction to your Rule database, you must enter the following information in the fields provided:

Frequency of the distribution

This is a frequency parameter of your choice, as discussed in “How propel Works” on page 13.

A Name for the rule

Every rule must have a name.

File query parameters

This must be a query formed according to the rules described in “Additional Database Query Techniques” on page 47.

Host query parameters

This must be a query formed according to the rules described in “Additional Database Query Techniques” on page 47.

Once you have entered the rule’s information, press the *Add* button by placing the cursor over the button and pressing the left mouse button. You see the information appear in the window in a slightly different format. This tells you that your entry has been accepted into the database.

The *ruleedit* window with a prepared entry looks something like the window in Figure 1-13.

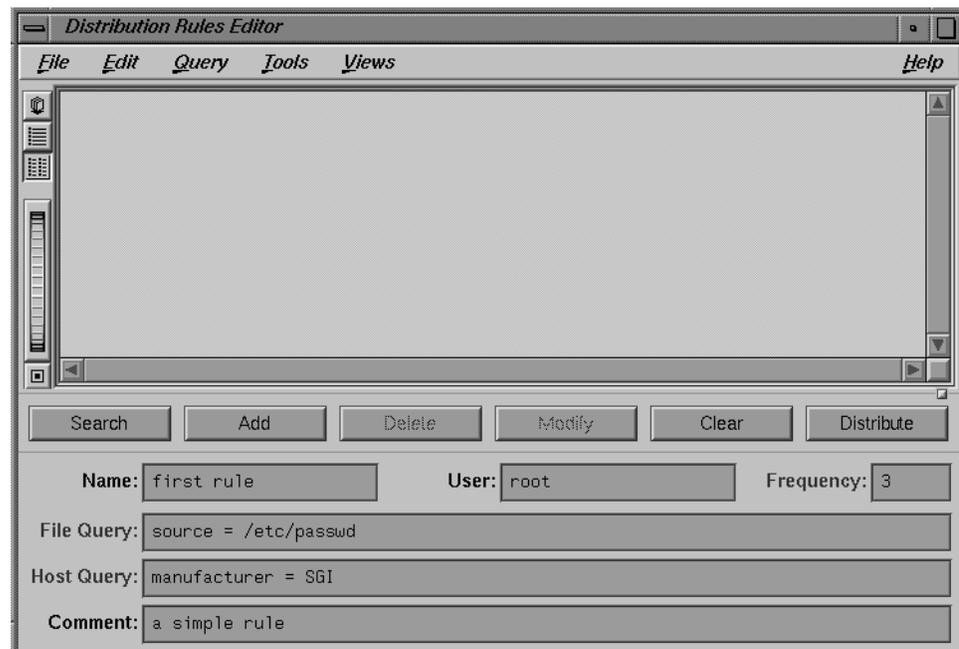


Figure 1-13 A Prepared Distribution Rule

Updating a Rule Database Entry

To change an entry, click the entry in the window with the left mouse button to highlight it. The information you entered appears in the fields in the lower part of the screen. Change these entries as you wish and then press the *modify* button.

Querying the Rule Database

To query the Rule database, begin by using the mouse to press the *Clear* button. This clears the fields of all information. Then enter your query words any field or combination of fields. Then use the mouse to press the *Search* button or press **<Enter>**. All rules in your database that match the query parameters are displayed in the window.

The User Database

The User database contains all the necessary information about the users on your network. For more information on users and groups and the contents of these fields, see the *IRIX Admin System Configuration Site and Server Administration Guide*.

Records in the User database have the following fields:

User ID	This field contains the user's unique UID number. UID numbers are integers from 0 to 65535. This field must be filled for an entry to be accepted. See the <i>IRIX Advanced Site and Server Administration Guide</i> for more information on assigning User ID numbers if you are unsure about this field.
Login	This field contains the login name of the user. This login name must be unique within your network domain.
Password	This field contains the user's encrypted password. By using the right mouse button, you can select an option to allow you to set the password in clear text. The password you set will be encrypted and the encrypted text will be placed in the field. You can also use the <i>passwd(1)</i> command and copy the encrypted string from the <i>/etc/passwd</i> file. Be certain to enter the encryption string exactly, or the password will not function correctly.
Shell	This field contains the user's default login shell. This is typically <i>/bin/sh</i> or <i>/bin/csh</i> .

Server	This field contains the name of the user's home system. This home system should also be the user's mail server.
Name	This field contains the user's full name. This field is required for an entry to be accepted.
Groups	This field contains the names of any user groups to which the user belongs. Multiple groups may be specified in a space-separated list. The first group listed in the user's primary group. This field is required for an entry to be accepted.
Home	This field contains the pathname of a user's home directory. (For example, this field might contain the pathname: <i>/usr/people/armistead</i>).
Projects	This field contains the names of any projects the user may be associated with. Multiple projects may be specified in a space-separated list. This field is provided as a convenient mechanism for making collections of users through queries. You may place any group-identifying tags in this field if you choose.
Comment	This field is available for any purpose.

Editing the User Database

This database contains the names and vital information of the users on your network. The command */usr/propel/bin/useredit* invokes this tool, and you see the following screen.

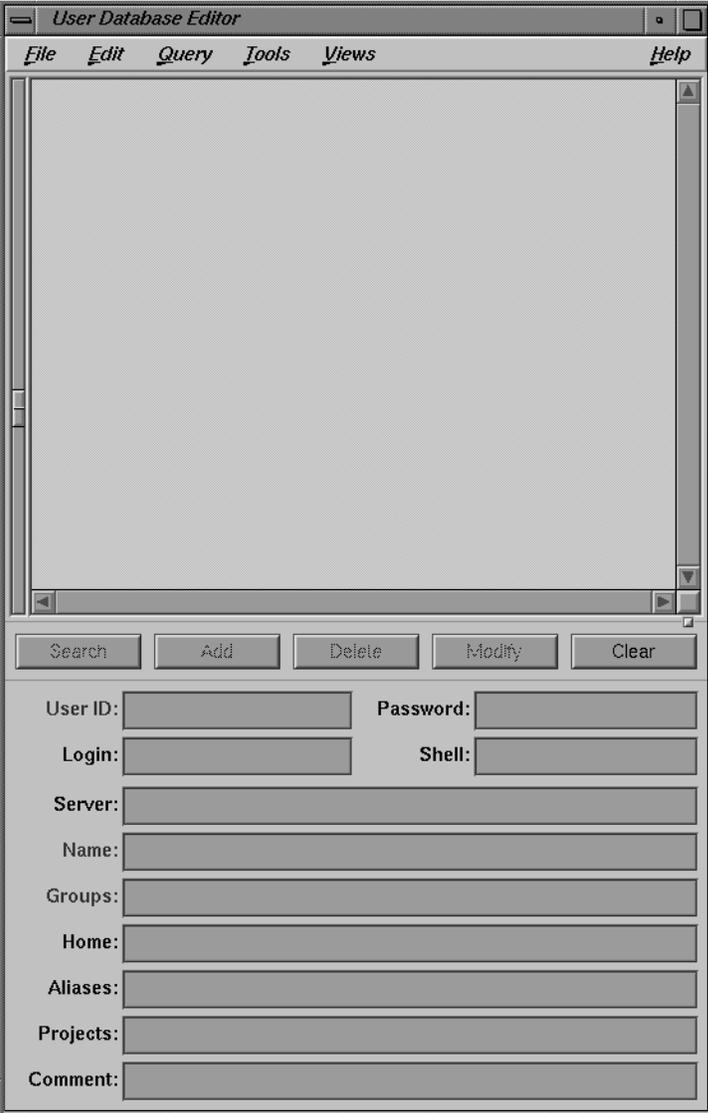


Figure 1-14 The User Database Editor

Adding a User to the User Database

To use the *useredit* tool to add a user to your User database, you must enter the following information in the fields provided:

- User ID
- Name
- Group (at least one)

The remaining fields are optional, but you should fill them in if at all possible for use in user queries. Any record associated with an actual user who will be logging in should be fully filled out. The minimal requirements allow you to reserve UID numbers without actually allocating them to specific users. It is sometimes useful to reserve blocks of UID numbers in advance if you wish to create future users in a certain group with consecutive UID numbers.

Once you have entered the user's information, press the *Add* button by placing the cursor over the button and pressing the left mouse button. You see the information appear in the display window in a slightly different format. This tells you your entry has been accepted into the database.

Updating a User Entry

To change this entry, clicking on the entry in the display window with the left mouse button highlights it, and the information you entered appears in the fields in the lower part of the screen. Change these entries as you wish and then press the *Modify* button.

Querying the User Database

For an example of using the User database fields as query points, you can make a query to the database to find all users that are part of the group "networking" and which have the comment "member of technical staff" Begin by using the mouse to press the *Clear* button. This clears the fields of all information. Then enter your query parameters in the appropriate fields. Then use the mouse to press the *Search* button. Records of all users in your database who match those two parameters will be displayed in the window.

The Group Database

Records in the Group database have the following fields:

Name	The name of the group of users. Make this name unique from any user or host name.
Group ID Number	The unique GID number associated with this group.
Query	The query parameters to define the group. You can query on any field in the User database. The field is made up of a list of typed fields separated by newlines (<Enter> keystrokes). The first word on each line is the type of query that the line represents. The available types are <i>username</i> , <i>userquery</i> , and <i>groupname</i> .
Comment	Any notes you wish to make about the group.

Editing the Group Database

The *groupedit* utility allows you to define groups of users for *propel*. A group of users is simply a selection of users that you have defined. Usually, groups are made of users with some common factor, such as a common project or organization. Since you specify the attributes of a user by using the *useredit* utility, you can prepare users to be part of a group when you enter them in your User database. For example, you can enter several users in your User database with the comment “advanced engineering lab” in the comment field. Then, when you use *groupedit* to create a group, you specify the “advanced engineering lab” comment as the query field when creating the group. The syntax for these queries is described in “Additional Database Query Techniques” on page 47. All users with that comment become part of the group. The names of individual users, User database queries, and other groups that define the new group are stored in the Group database.

Once defined, the *groupedit* utility stores the parameters of the group in a database. Enter the command:

```
/usr/propel/bin/groupedit
```

you see the following window, shown in Figure 1-15:

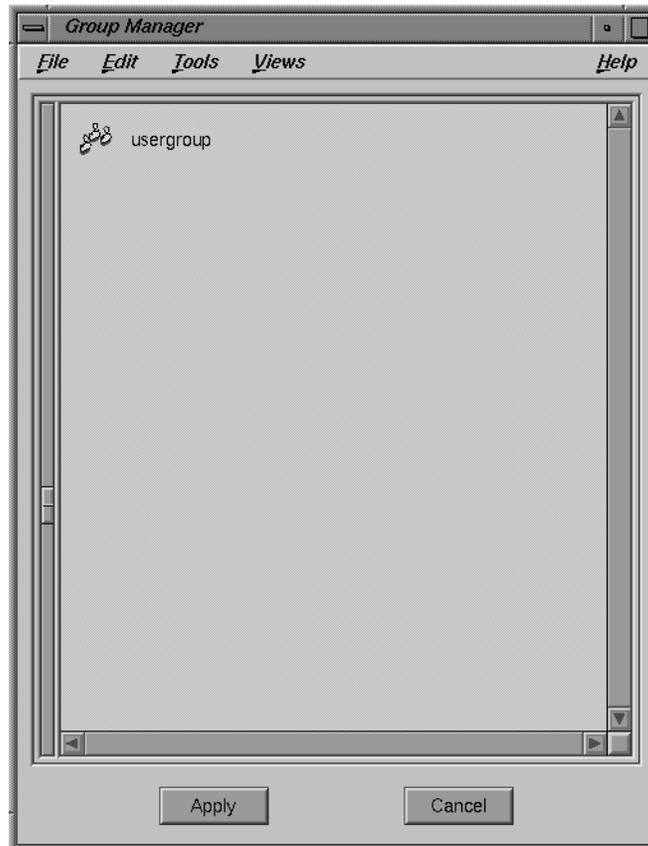


Figure 1-15 The Group Manager

When you bring up the Group Manager window after your database is populated, you see an icon and a name for each group in the database.

Adding a Group of Users to the Group Database

To use the *groupedit* tool to add a user group to your Group database, place the mouse cursor inside the *groupedit* window and use the right button of the mouse to bring up the popup menu. There are several options:

- Help Brings up a help card.
- Add Group Brings up the Add Group window.
- Add User Adds a user to an existing group.
- Remove Selected
 Removes the selected group.
- Edit Selected Users
 Brings up the *useredit* tool. To select a user, you must have a group displayed in a group display window.
- Preview Shows you the list of users currently in a group. All User database queries and other groups that are part of the current group are fully expanded.

Select the Add Group option to create a new group of users. When you select this option, you see the new window, shown in Figure 1-16.



Figure 1-16 The Add Group Window

You must enter a name for the group, then use the left mouse button to press the *Accept* or *Apply* button to create your new group. The *Apply* button creates your new group, and retains the Add Group window, so you can continue to create more groups conveniently. The *Accept* button creates your group and removes the Add Group window. The *Cancel* button cancels the operation without creating the new group.

Once you have created the group, you must specify users to be part of the group. You should notice that a new icon has appeared in the *groupedit* window. If you have previously used the *add_user* script, (described in “Populating the User and Group Databases”) your Group Manager window will already show many icons representing your existing user group. If you make two new groups and call one “irixpro” and the other “user” the window looks similar to Figure 1-17.

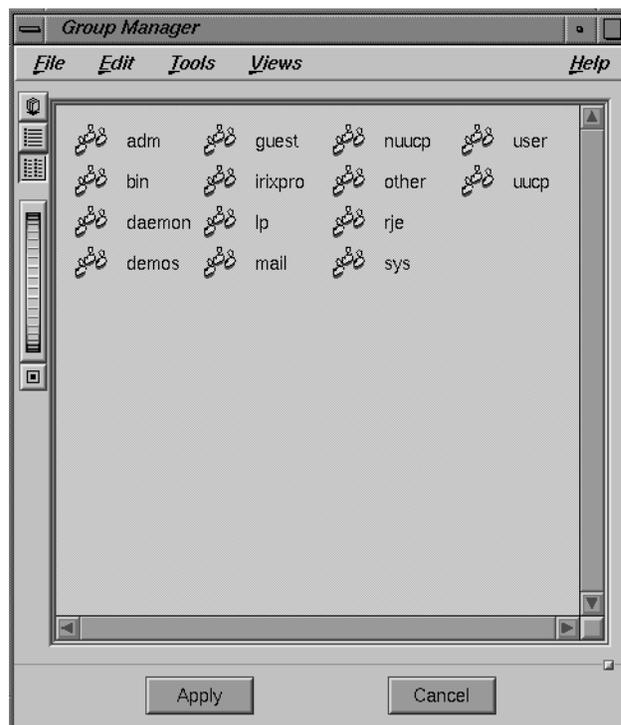


Figure 1-17 A Populated Group Manager Window

To add a list of users, a list of groups, a User database query, or any combination of these things to your new group, double-click the left mouse button with the mouse cursor over the icon for the new group. You see a new window that looks something like Figure 1-18.

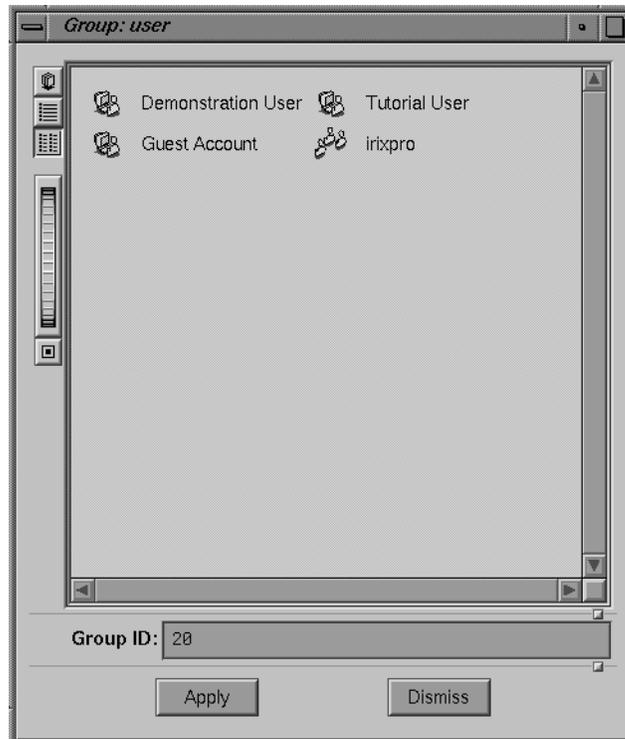


Figure 1-18 A Group Display Window

Now use the right mouse button to bring up the popup menu, and select either the Add User or Add Group options to populate your group.

The Add User option brings up the Add User window, shown in Figure 1-19, with the entire list of users from your User database.



Figure 1-19 The Add User Window

To place a user in your new group, place the mouse cursor over the name of the user you wish to select and double-click the left mouse button. When you are finished selecting individual users, click the *Accept* button to secure your changes and remove the Add User window.

The Add Group option brings up the Add Group window, previously shown in Figure 1-16. All your currently configured groups are shown in this window. As with the Add User window, double-click the name of a group you wish to add to your new group and complete your changes by pressing the *Accept* button.

Once you have entered all the information and created your groups, press the *Apply* button in the Group Manager window to place your groups in the Group database.

Updating a Group Database Entry

To change a Group database entry, click the icon with the left mouse button to highlight it, and the Group Display window for that group appears. Change these entries as you

would if you were making a new group, then click the *Apply* button to finalize your changes.

Additional Database Query Techniques

The *propel* databases can be queried using query language rules and glob expressions. The rules regarding glob expressions are based on the Bourne shell glob expressions. See the section of the *sh(1)* reference page titled *File Name Generation* for additional documentation on glob expressions.

There are two basic ways to query from the graphical utilities: direct and indirect queries.

Direct queries are queries to the current database made through the editor for that database. For example, to query the Host database through the *hostedit* utility, you clear the screen with the *Clear* button on the utility, then enter your query parameters in the appropriate fields and press the *Query* button using the mouse.

Indirect queries are queries to a database other than the current database. For example, an indirect query would happen if you are editing the Rule database and you wish to query the Host database for a list of hosts to use in the rule you are creating.

The principal difference between the two forms of database query is that in the direct query, you simply fill in the desired parameter in the provided field and press the *Query* button using your mouse. To make an indirect query, you must fill in the field name from the database you are querying and also fill in the desired parameter value. For example, a simple indirect query in the *host query* field of the Rule database has the form:

fieldname = value

Any propel query menu allows you to form a query in this manner, and all query fields have an option available from the popup menu (brought up by pressing the rightmost button on the mouse while the cursor is within the query window) to expand the query immediately.

If you use this form to query for a host named *pickett*, you would form the actual query like this:

```
hostname = pickett
```

You cannot immediately see the results of your query on an indirect query unless you explicitly expand it. The results are not printed in the display window of the database editor. The query takes place only when *propel* executes and performs the distribution based on your query.

The advantage to this method is that you do not need to rebuild your databases if you add a new file or host that will be subject to a query. For example, if you have set a distribution to be sent to all hosts with the model of "Indy" on a regular basis, you do not need to remake the distribution rule if you add another "Indy" host. The new host is automatically selected by the existing query. If you wish to check to see if your query parameters will produce the desired results, use the editor for the database you are querying indirectly or the *dbredit* command to make a test query using the same parameters.

A glob expression in this type of query has the format:

fieldname ~ expression

For example, an actual query would be similar to this:

```
model ~ indigo*
```

or similar to this:

```
os ~ "IRIX 5.*"
```

You can use several query parameters at once on both direct and indirect queries. In a direct query, simply fill in as many fields as you like before you press the *Query* button on your database editor. For an indirect query, you must form the query according to query language rules. For example, to query for all hosts whose manufacturer is "SGI" or whose model name begins with the characters "Ind" use the following query:

```
manufacturer = SGI, model ~ Ind*
```

With the above query, you have selected all hosts whose manufacturer is listed as "SGI" as well as all hosts whose model name begins with "Ind." This is known as a logical "OR" query.

To select only those hosts whose manufacturer is listed as "SGI" and whose model is listed as "Indy," use the following logical "AND" query:

```
manufacturer = SGI && model = Indy
```

The difference is that the second query selects only those hosts that fulfill both requirements, while the first query selects those hosts that fulfill either requirement.

You can further expand your queries by incorporating more parameters. For example, to select those hosts whose manufacturer is "SGI" and whose model is "Indy," and also the host named "callahan," and any host with the comment "Universal," use the following query:

```
manufacturer = SGI && model = Indy, hostname = callahan,  
comment = universal
```

Query Keywords and Glob Expressions

The query rules follow simple glob expression syntax. The following keywords and operators are recognized:

all The **all** keyword returns all elements in the database. For example, the Host database query

```
all && !labmachine
```

returns all the hostnames in the database except *labmachine*.

to The **to** keyword allows you to specify a range of records. For example, the Host database query

```
iplist = 192.27.3.0 to 192.27.3.255
```

returns all addresses in the specified range. In the above example, all hosts on the 3 subnet would be returned.

= The equal sign operator is the most basic operator. Given alone, it indicates that the expression following describes the desired query parameters.

! The exclamation point operator indicates a logical "NOT" operation. This operator negates the usual meaning of a query parameter. For example, if you want to query for all SGI manufactured hosts except the host "bragg," use the query:

```
manufacturer = SGI && ! hostname = bragg
```

- !=** The exclamation point negation operator followed by an equal sign indicates “everything not equal to the following expression.” For example, the query

```
manufacturer != SGI
```

selects all manufacturers except SGI.
- >** The Greater-Than operator selects all values greater than the given expression.
- >=** The Greater-Than or Equal To operator selects all values greater than or equal to the given expression.
- <** The Less-Than operator selects all values less than the given expression.
- <=** The Less-Than or Equal To operator selects all values less than or equal to the given expression.
- &&** The double ampersand operator indicates a logical “AND” operation. It is used to indicate that both parameters in a query must be satisfied to create a match. For example, to select those hosts whose manufacturer is “SGI” and whose model is “Onyx,” use the following query:

```
manufacturer = SGI && model = Onyx
```

You may use as many logical “AND” operators together as you wish, and the meaning remains the same. All query parameters put together with “AND” operators must be satisfied to create a match.
- ||** The double pipe operator indicates a logical “OR” operation. This means that only one query parameter needs to be satisfied to create a match. For example, the query:

```
manufacturer = SGI || owner = armistead
```

matches all hosts whose manufacturer is “SGI” or that are owned by “armistead” regardless of their manufacturer. You may use as many logical “OR” operators together as you wish and the meaning remains the same. Only one query parameter in an “OR” string must be satisfied to create a match.

- () Parentheses enclose expressions that need to be evaluated separately. For example, suppose you wish to evaluate a query to find all hosts whose model is "Indy" except those whose OS Release is "5.2," and then add a query for all hosts whose model is "Indigo." In this case, you would use parentheses to force the first parameters to be evaluated before adding the last parameter. The syntax for this query would be:

```
(model = Indy && ! os = 5.2), model = indigo
```

Query Shortcuts

Query shortcuts exist for each database. When querying the Host database, the following shortcut rules apply:

- A single word, such as "upstairs," is assumed to indicate the name of a collection. Thus, "upstairs" is interpreted as equal to the query "collect:name = upstairs."
- A string of letter characters with periods interspersed is assumed to indicate an individual hostname. Thus, "gordon.eng.biz.com" is interpreted as equal to the query "hostname = gordon && domain = eng.biz.com."
- A string of numeral characters with periods interspersed is assumed to indicate an IP address. Thus, "192.27.0.1" is interpreted as equal to the query "iplist = 192.27.0.1."
- A field name listed by itself matches all records where that field has an entry. Thus, "owner" returns all records where an owner is listed.

When querying the File database, the following shortcut rules apply:

- A single word, such as "newfiles," is assumed to indicate the name of a package of files. Thus, "newfiles" is interpreted as equal to the query "package = newfiles."
- A string of letter characters with slashes (/) interspersed is assumed to indicate a source file pathname. Thus, "/var/propel/data/etc/hosts" is interpreted as equal to the query "source = /var/propel/data/etc/hosts."
- A field name listed by itself matches all records where that field has an entry. Thus, "source" returns all records where a source filename is listed.

When querying the Collection database, the following shortcut rules apply:

- A single word, such as "upstairs," is assumed to indicate the name of a collection. Thus, "upstairs" is interpreted as equal to the query "name = upstairs."

- A field name listed by itself matches all records where that field has an entry. Thus, “**name**” returns all records where a name is listed.

When querying the Rule database, the following shortcut rules apply:

- A single word, such as “**walnuts**,” is assumed to indicate the name of a rule. Thus, “**walnuts**” is interpreted as equal to the query “**name = walnuts**.”
- A field name listed by itself matches all records where that field has an entry. Thus, “**frequency**” returns all records where a frequency is listed.

Using the dbredit Utility

The *dbredit(1)* utility is a command-line interface to the *propel* databases designed for use in server environments where the graphical utilities cannot be used. All functions that are performed by the graphical utilities can also be performed with *dbredit*. Complete information on the *dbredit* command is provided in the *dbredit(1)* reference page. The basic syntax of *dbredit* is:

dbredit *database command information*

where *database* is the name of the *propel* database to be manipulated or queried, *command* is an operation corresponding to the operations performed through the graphical editors, and *information* is the information to be entered into the database or the query parameters.

The *database* variable can be one of *host*, *file*, *rules*, or *collect*. The *command* variable can be one of *classes*, *init*, *list*, *insert*, *query*, *delete*, or *update*. The *information* variable is a set of

arguments specific to each *command*. The arguments for each *command* are listed below:

classes This command prints information about all databases currently configured.

init This command initializes the named database.

list Takes no arguments. This command returns all records in the database. For example, to see your entire Host database, use the command:

```
dbredit host list
```

You see output similar to the following:

```
{hostname {sherman}} {manufacturer {SGI}} {os {IRIX 5.3}}
{domain {enr.com}} {model {IRIS Indigo2}} {iplist
{127.0.0.1 192.999.888.7}} {maclist {{
08:00:69:07:94:D7}} {mxlist {{{}}}

{hostname {maguire}} {manufacturer {SGI}} {os {IRIX 5.3}}
{domain {wpd.sgi.com}} {model {IRIS Indy}} {iplist
{192.555.444.2}} {maclist {08:00:69:06:C1:95}} {mxlist
{{{}}}
```

There is one output entry per entry in your database, and each field in the entry is listed along with its value, if any. For example, in the above listed output, the first field reads:

```
{hostname {sherman}}
```

The first word is the field name, and the second word (inside braces) is the field value. The whole field statement is further enclosed in braces. Where there is no value for the field, only the field name and empty braces are displayed, as follows:

```
{mxlist {{{}}}
```

add Takes a list of "attribute = value" arguments that correspond to the fields in the appropriate database and returns the entityID number of the new record. For example, to add a typical file record, you might use the following command:

```
dbredit file add source=/tmp/localfile \
destination=/tmp/remotefile
```

search Takes a list of queries or entityID numbers, and returns the matching records. For example, to search for all systems manufactured by Silicon Graphics, use the command:

```
dbredit host query manufacturer=SGI
```

delete Takes a list of query strings or entity ID numbers, and returns the records that have been deleted. For example, to delete the records for all systems manufactured by "Tredegar," use the command:

```
dbredit host delete manufacturer=Tredegar
```

<i>modify</i>	<p>Takes a list of queries or entity ID numbers, and returns the entity ID numbers of the updated records. For example, to change all of the operating system version strings for systems running "IRIX 5.2" to "IRIX 5.3" use the command:</p> <pre>dbredit host modify os="IRIX 5.2" os="IRIX 5.3"</pre>
<i>destroy</i>	<p>Removes all values of a named field or database. For example, the command</p> <pre>dbredit host destroy owner</pre> <p>removes all values of the "owner" field from the Host database. The command</p> <pre>dbredit host destroy</pre> <p>removes the entire Host database. After a destroy operation, the database may be reinitialized or simply repopulated.</p>
<i>names</i>	<p>Returns the complete list of fields for the given database. For example, the command</p> <pre>dbredit host names</pre> <p>returns all the field names in the Host database. Given with no database name, the command returns a list of the known databases. For example, the command:</p> <pre>dbredit names</pre> <p>returns the names of all known databases.</p>

All rules about query structure and glob expressions that apply to the graphical database editing utilities also apply to *dbredit*. If you have trouble creating the correct syntax, try issuing a **dbredit names** command and entering the field names exactly as they are displayed.

Running propel

The *propel* program can be executed automatically by the *cron* process, or from a shell command at your convenience. The following sections describe the two ways of running *propel*.

Running propel From a Shell Command

There is a shell command, *runpropel(1M)*, available to force a distribution to a host (or collection of hosts) immediately. Command line arguments are available to specify a direct push or a periodic push of any standard frequency. For complete information on *runpropel(1M)*, see the *runpropel(1M)* reference page.

Running propel Through cron(1M)

The *runpropel* command can be run by *cron(1M)* regularly. Each time *cron* executes *runpropel*, the program reads and interprets the various databases and then generates or modifies all necessary files.

An example entry from a distribution system's *crontabs* file would be:

```
runpropel -f week
```

The frequency of the push is configurable at any time by changing the command in your *crontabs* file and restarting *cron*. For complete information on *cron*, see the *cron(1M)* reference page, as well as the section titled "Automating Tasks with *at(1)*, *batch(1)*, and *cron(1M)*" in the *IRIX Advanced Site and Server Administration Guide*.

The propel File Generation System

The file generation system consists of a directory of scripts that are run by *propel*, if needed, before the actual distribution takes place. If an entry in the File database has a program listed in the "generator" field, *propel* runs the generation program and takes the output from that program to replace the contents of the existing file. For example, you can create an entry in the File database in which the destination file is */etc/hosts*, and the "generator" field lists the */usr/propel/sample/make_hosts* command. In this case, the *make_hosts* command builds an */etc/hosts* file from the entries of the *propel* Host database, and then distributes the resulting file to all client hosts called for in the distribution rule, installing the new file as */etc/hosts*.

Some utilities to generate files are provided with *propel*, though you can always create your own file-generation utilities. By using the utilities provided with *propel*, and by propagating the resulting files, you can be sure that your entire network maintains consistent system configuration files. There are several file-generation scripts shipped with *propel* in the */usr/propel/sample* directory:

- make_hosts* Generates the */etc/hosts* file from the *Host database*.
- make_ethers* Generates the */etc/ethers* file from the *Host database*.
- make_fwd_dns* Generates the forward maps for *named(1M)* to be distributed to the *named* master.
- make_rev_dns* Generates the reverse maps for *named(1M)* to be distributed to the *named* master.

Index

A

add_hosts command, 5, 6
administration, system
 documentation, xii-xiii

B

batch_update command, 6

C

coledit
 adding a record, 23, 42
 command, 12, 23, 40
 picture, 23
 updating records, 29, 46
Collection database, 22, 40
 description, 4
 editing, 22, 40
commands
 add_hosts, 6
 batch_update, 6
 coledit, 12, 23, 40
 dbredit, 13, 52
 fileedit, 12
 find_hosts, 6, 7
 groupedit, 12
 host_info, 6
 host_parser, 5

hostedit, 12, 19
lm_tcp, 9
make_hosts, 55
propel, 12
rdist, 9
ruleedit, 12, 34
runpropel, 14, 55
snmp_push, 7
update_hosts, 6
useredit, 12
conventions, typographical, xv
customizing
 propel, 17

D

databases
 Collection, 22, 40
 File, 29
 Host, 17
 Rule, 33
dbredit command, 13, 52
distributing software with propel, 15
Distribution Rule database, 33
documentation conventions, xvi

E

entry fields, using, xix

F

File database, 29
 editing, 30
 loading, 8
fileedit
 adding a record, 31
 command, 12
 picture, 30
 querying the database, 33
 updating records, 33
file prompter windows, using, xx
find_hosts command, 6, 7

G

groupedit
 command, 12

H

help
 reference, xiii
host_info command, 6
host_parser command, 5
Host database, 17
 description, 3
 initialization, 6
 initializing, 5
 loading, 5
 populating, 6
hostedit
 command, 12, 19
 insterting records, 21
 parsing system files, 22
 picture, 19
 querying, 21
 updating records, 21

host management with propel, 16

I

installation
 propel, 2
IRIX administration
 documentation, xii-xiii

L

lm_tcp command, 9
localizing propel, 17
LOGERRS variable, and rdist, 10

M

make_hosts command, 55
man command, xiii
man pages, xiii

O

options buttons, using, xx

P

propel
 and cron, 54
 and LOGERRS variable, 10
 and rhosts file, 10
 and trusted hosts, 10
 command, 12
 customizing, 17
 distributing software, 15

file generation system, 55
host management, 16
installation, 2
installing shareware, 16
running, 54
scheduling, 13

Q

querying
 advanced techniques, 47
 keywords and regular expressions, 49
 propel databases, 47
 Rule database, 36
 shortcuts, 51

R

rdist
 LOGERRS variable, 10
 source for heterogeneous networks, 9
 utility, 9
rhosts file and propel, 10
Rule database, 33
ruleedit
 adding a record, 34
 command, 12, 34
 picture, 34
 updating records, 36
runpropel command, 14, 55

S

scheduling propel, 13
scroll bars, using, xviii
shareware, distributing with propel, 16
SNMP

 description, 10
snmp_push command, 7
snmpd
 description, 10
 installation on remote hosts, 7
 running on all systems, 11
 SNMP agent, 11
system administration
 documentation, xii-xiii

T

trusted hosts and propel, 10
typographical conventions, xv

U

update_hosts command, 6
useredit
 command, 12
user interface operations, xviii
user interface terms used in this guide, xvi, xviii

W

window terms used in this guide, xvi, xviii

Tell Us About This Manual

As a user of Silicon Graphics products, you can help us to better understand your needs and to improve the quality of our documentation.

Any information that you provide will be useful. Here is a list of suggested topics:

- General impression of the document
- Omission of material that you expected to find
- Technical errors
- Relevance of the material to the job you had to do
- Quality of the printing and binding

Please send the title and part number of the document with your comments. The part number for this document is 007-3272-001.

Thank you!

Three Ways to Reach Us

- To send your comments by **electronic mail**, use either of these addresses:
 - On the Internet: techpubs@sgi.com
 - For UUCP mail (through any backbone site): *[your_site]!sgi!techpubs*
- To **fax** your comments (or annotated copies of manual pages), use this fax number: 415-965-0964
- To send your comments by **traditional mail**, use this address:

Technical Publications
Silicon Graphics, Inc.
2011 North Shoreline Boulevard, M/S 535
Mountain View, California 94043-1389