System Configuration Repository User's Reference

HP-UX/HP 9000 Computers

Edition 1.2

April 2001

Printed in: USA

Legal Notices

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty

A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in sub-paragraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

Hewlett-Packard Company 3000 Hanover Street Palo Alto, CA 94304 U.S.A.

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Trademark Acknowledgments

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

Contents

1. Introduction

2. SCR Overview

Installing SCR 13
Using SCR. 14
Configuring SCR. 15
Registering Managed Nodes. 15
Setting the Data Collection Schedule 16
Collecting Configuration Data On Demand 17
Checking Data Collection Status. 18
Viewing Configuration Information on a Node. 19
Tracking Configuration Changes with SCR 20
Checking Node Differences 21
Using Tags to Simplify Node Management. 23
Using Filters. 25
Simple Filter Example 26
Standard Filters 28
Using the Probe Filter 29
Additional Uses of SCR 30
Inventory Tracking 30
Change Management 31
Disaster Recovery 32

SCR Reference (man Pages) 33
scr (5) 34
scrconfig (1M) 43
scrdelete (1M) 48
scrdiff (1M) 50
scrfilter (1M) 53
scrhist (1M) 64
scrstatus (1M) 66
scrtag (1M) 69
scrupdate (1M) 72
scrviewer (1M) 74
scrlog_viewer (1M) 77

A. SCR File Set

Contents

About this Manual

Printing History

The printing date will change when a new edition is printed. Minor changes may be made at reprint without changing the printing date. The manual part number will change when extensive changes are made.

Manual updates may be issued between editions to correct errors or document product changes. To ensure that you receive these updates or new editions, see your HP sales representative for details.

April 2001...Edition 1.2.

April 2000...Edition 1.1.

How to Use this Guide

This is a technical reference for the ServiceControl Manager. It deals with the concepts and technical details necessary to create a managed cluster using ServiceControl Manager and to efficiently manage the systems using the ServiceControl tools.

This guide contains the following information:

- *Chapter 1 , "Introduction,"* provides an introduction to this manual.
- Chapter 2 , "SCR Overview," provides an overview of the System Configuration Repository.
- Chapter , "SCR Reference (man Pages)," includes the man pages for the System Configuration Repository.

  Appendix A , "SCR File Set," describes the file set for the System Configuration Repository.

Comments about this manual (no technical questions, please) can be directed to: editor@fc.hp.com. Please consult your HP service and support representative for technical support questions. Thanks.

Typographic Conventions

This guide uses the following typographic conventions:

Boldface	Words defined for the first time appear in boldface. For example, an **argument** is the part of a command line that indicates what file or directory the command is to act on.
`Computer`	Computer font indicates literal items displayed by the computer. For example: `file not found`
`User input`	Bold, computer text indicates literal items that you type. For example: **`cd`**
Italics	Manual titles and emphasized words appear in italics, as do values that you supply. For example, in the command below you would substitute an actual directory name (such as `mydir`) for *directory_name*. `cd` *directory_name*
Enter	Text in a bold, sans serif font denotes a keyboard key. A notation like **CTRL**+**Q** indicates that you should hold the control key down, then press **Q**.
Softkey	Select an on-screen item or a corresponding softkey. For example, `Help` shown at the bottom left side of the screen means that pressing the softkey corresponding to that position on the screen (f1) will cause a help screen to be displayed.

1 Introduction

This manual describes the features of Hewlett Packard's System Configuration Repository (SCR) for HP-UX and how to use them. Although powerful, SCR is simple to use, if you have a basic understanding of the HP-UX command syntax.

- Chapter 2 describes how to use SCR commands to configure the program for operation. This is all you need to do to begin tracking changes in your system quickly and easily.
- Chapter 3 describes the full capabilities of all 10 SCR commands.

The best way to learn SCR's capabilities is to experiment with the commands. SCR does not change any system configurations. Using a command incorrectly may give you some error messages or strange information, but it will not harm your system.

SCR Features

SCR provides the following time-saving features:

- SCR can extract a static "snapshot" of a system's configuration by request or by schedule, and store it in the persistent repository. A snapshot is a record of a system's configuration at a specific time.
- SCR snapshots stored in the repository can be compared with each other to identify any differences.
- SCR configuration snapshots include all relevant configuration information included in the Management Information Format (MIF) for the Desktop Management Interface (DMI). See the DMI User's Reference for information about DMI and the MIF.

SCR shows you quickly what is going on in a system. As root user, you can enter a command to learn the current status of a particular system or an entire set of systems. With options to that command, you can learn what changes have taken place recently. If desired, these commands will execute according to a schedule and automatically generate reports. By monitoring changes in a set of similarly configured systems, SCR can give you advance warning of potential problems.

Using SCR, you can retrieve a snapshot and compare it to a previous snapshot. This may reveal things that are going wrong in a system but not generating error messages. SCR uses DMI programming routines to gather information from systems. Be sure DMI is running and managing each node for which you want a snapshot. See the DMI 2.0 User's Reference for DMI installation and configuration instructions.

2 SCR Overview

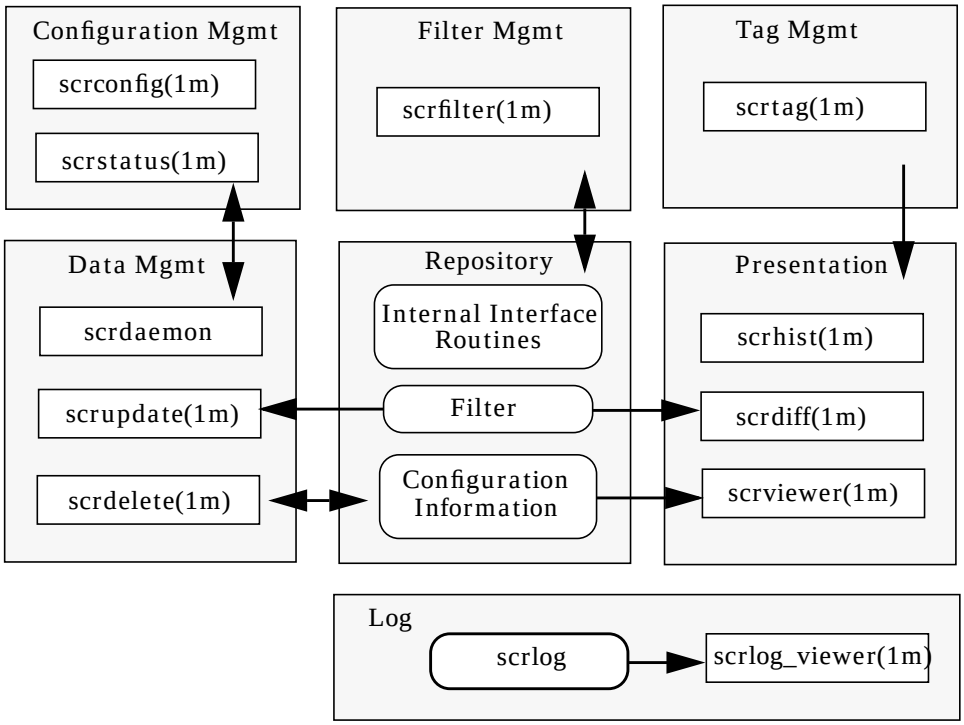
This chapter describes SCR and how it works. Then it tells you how to install SCR and how to configure it for first use. With this information, you can be tracking system changes and viewing reports about system nodes with SCR within a short time.

SCR consists of seven components:

- Configuration Management
- Data Management
- Filter Management
- Tag Management
- Presentation
- Repository Management
- Log Components.

The Repository component provides the internal software interface. The other components provide the external user interface to SCR. Figure 2.1 shows the SCR Component structure.

Figure 2-1 SCR Component Structure



The Configuration Management component provides the `scrconfig` and `scrstatus` commands. `Scrconfig` maintains the configuration data collection schedule and managed node list. `Scrstatus` generates a status summary report of data collection activity.

The Data Management component provides the `scrdaemon`, `scrupdate`, and `scrdelete` commands. `Scrupdate` collects configuration data from a managed node through the DMI interface and stores data in the repository. `Scrdelete` removes specified or expired configuration data from the Repository. `Scrdaemon` invokes `scrupdate` based on the data collection schedule maintained by Configuration Management component.

The Filter Management component provides the `scrfilter` command to refer to and update view filters. View filters are stored in the SCR Repository.

The Presentation component provides the `scrhist`, `scrdiff`, and `scrviewer` commands. These commands retrieve stored configuration information and display it in human readable format.

The Tag Management component provides user friendly naming capability for a snapshot. Using `scrtag`, you can change the name “200102010005” for a snapshot to “StableSystem”. You can then use StableSystem instead of 200102010005 in commands to the Presentation components `scrdiff` and `scrviewer`.

The Log Component provides the `scrlog_viewer` command. This command displays all SCR activities that have been recorded in a log file with a graphical user interface.

Installing SCR

SCR is part of the SCR+DMI bundle. Install SCR+DMI to use SCR. SCR can only work on system nodes that have been configured to be managed by DMI.

See the DMI User's Reference document for an explanation of DMI installation, concepts, and operations.

When SCR+DMI has been installed and the system configured, you can begin using SCR commands to monitor your system, as described in the next section.

Using SCR

SCR has two types of management systems:

Table 2-1 SCR Management Systems

System Type	Description
Central Management Server (CMS)	An HP-UX system that maintains SCR
Managed Nodes	Other HP-UX systems to be managed by the CMS.

SCR can perform scheduled and on-demand configuration data collection on the CMS and all managed nodes.

For configuration data collection and management of system nodes, DMI must be managing the nodes. You must include the CMS IP address in that server's `/var/dmi/dmiMachines` file. You also must add the CMS IP address in each managed node's `/var/dmi/dmiMachines` file to allow access from the CMS.

The next step is to configure SCR to perform the data collection and reporting you want, as described in the next section.

Configuring SCR

SCR must be configured to perform its data gathering and storing tasks. Use the **scrconfig** command to specify SCR tasks for each managed node.

Registering Managed Nodes

To begin, each managed node must be registered using the **scrconfig** command. To register a managed node named *elsa*, type the following command on your management server:

```
# scrconfig –n +elsa
```

The `-n` option to the scrconfig command applies the command to a particular node. The `+` sign tells SCR to add the node. A `-` sign would remove the node. This node is named `elsa`.

NOTE The request to register a managed node may fail if DMI is not properly configured on the node or disallows access to the node from the CMS.

Repeat this operation for each system node that you want to track. You can add multiple nodes with one command:

```
# scrconfig –n +node1 +node2 +node3
```

Substitute your node names for *node1*, etc. above. You can add as many nodes as you want in one command. As the Repository can become quite large, we recommend a maximum of 20 to 50 nodes per CMS, depending on your CMS resources.

Setting the Data Collection Schedule

The next step is to set up a day and time for the first data collection on each registered node. SCR will collect data automatically at that day and time each week. It takes a "snapshot" of the node data and stores it in the repository.

Set the data collection schedule with the scrconfig command. For example, enter the following command to set scheduled data collection time at 12:05 Feb 7, 1999 for managed node elsa.

```
# scrconfig –n elsa –s 199902071205
```

As before, `-n` specifies a particular node that is named immediately afterward. The `-s` option tells SCR to collect data at the time that follows. Enter the time as shown above, with the full year digits, followed without spaces by the month, day, and time digits. Time uses the 24-hour clock.

To confirm that you entered the current schedule parameters correctly, enter the following command:

```
# scrconfig –n elsa –l
```

The `-l` option tells SCR to list the schedule details for `elsa` on stdout.

The result resembles the following example:

```
NODE    SCHEDULE    TIME        INTERVAL    EXPIRATION     TIMEOUT
elsa    02/07/1999  13:05 PST   1 day       3 months       5 minutes
```

This example shows that data will be collected from node `elsa` at 1:05 PM Pacific Standard Time on February 7, 1999, and will occur each day after that at the same time. Snapshots will be deleted automatically when they are three months old. (Snapshots that have been named with the scrtag command will not be deleted. See the `scrtag(1M)` man page in Chapter 3 for details.) The timeout parameter indicates that SCR will try to collect data for five minutes each time, before giving up. Data collection interval, expiration, and timeout parameters can be customized on the command line, as well. See the `scrconfig(1M)` man page or the description of the `scrconfig` command in Chapter 3 for details.

If you omit the `-n` option and node name, `scrconfig` lists schedule details for all registered nodes. A daemon invoked by the `init` process controls data collection.

You can disable automatic data collection with the command:

```
# scrconfig –s off
```

Collecting Configuration Data On Demand

The above steps set up automatic data collection on particular nodes at particular times. For immediate data collection, use the `scrupdate` command. For example, enter the following command to get a data snapshot of the managed node `ortrud`:

```
# scrupdate –n ortrud
```

You will see a message similar to the following:

```
Configuration data “ortrud:time” is collected.
```

Collection can take from 5 to 15 minutes (and sometimes longer), depending on the node size, network load, and how much data is collected. See Using the Probe Filter, Chapter 3, to learn how to control the amount of information that SCR collects from a node.

If configuration data for `ortrud` has changed since the last collection, the new data will be placed in the SCR repository. If configuration data has not changed, no data is stored.

Checking Data Collection Status

To check data collection status of all registered nodes, use the `scrstatus` command. This provides a summary report of the last, current, and scheduled data collection for all registered nodes, in time order.

```
# scrstatus
```

The report, printed to stdout, resembles the following example:

```
TIME  (START – STOP)              NODE      STATUS      DETAIL

02/06/199912:05 – 12:08 PST       elsa      Completed

02/06/199912:15 – 12:17 PST       ortrud    Warning     -   AcErr  -


02/07/199916:37 -         PST     ortrud    Executing


02/07/199900:05 -         PST     ortrud    Scheduled

02/07/199912:05 -         PST     elsa      Scheduled
```

This report appears in three blocks separated by vertical space. The first block above (the first two lines) reports the most recent data collection. The second block shows data collection being performed. The last block shows future scheduled data collection. Some of these blocks might be empty, especially when SCR is first installed.

Viewing Configuration Information on a Node

Use the scrviewer command to see configuration information on a node at a certain time. In the following example, we are using the "latest" tag to get the most recent collected data. SCR automatically assigns the latest collection time to the tag. The report tells us what that time is.

```
# scrviewer elsa:latest

elsa : 04/10/1999 13:05 PDT  (latest)

COMPONENT NAME                        VALUE

    GROUP NAME

        ATTRIBUTE NAME

HP Visualize Workstation

      :

"HP-UX Standard Groups Definition"

      :

    "General Information"

        [System Name]              elsa

        [System Location]          3000 Hanover St, Palo Alto, CA

      :
```

If we wanted data from an earlier snapshot, we would have to enter the time of the snapshot instead of the "latest" tag, or create a tag to represent that information. (See "Using Tags to Simplify Node Management" on page 23, for additional information about tags.)

Tracking Configuration Changes with SCR

Use the `scrhist` command to check the configuration collection history for managed nodes. This command reports a brief list with node, time and tag information. To generate a report for all system nodes, enter the following command:

```
# scrhist
```

This command generates a report that resembles the following example:

```
NODE      TIME                          ERR       TAG
elsa      12:05 02/07/1999 PST                    oldest
          12:05 02/14/1999 PST
          13:05 04/10/1999 PDT                    latest
ortrud    12:25 03/23/1999 PST                    oldest
          13:25 04/10/1999 PDT                    latest
```

If no time period is specified, as in this example, the report includes all configuration collections that have not expired. The SCR tag, oldest, stands for the oldest configuration update that has not expired.

SCR does not store a snapshot if there are no differences in the node configuration since the previous snapshot. You can use this feature to observe how often the configuration of certain nodes changes. If data has been automatically collected weekly, this report shows that node elsa's configuration changed between February 7 and February 14, 1999, but did not change again until April 10, 1999.

You can limit the scope of the list by time period and managed nodes. To limit the time period, enter one of the following lines:

```
# scrhist –p 19990101 19990226
```

```
# scrhist –p 19990401 latest
```

The `-p` option signifies that the digits that follow are a time period that limits the report.

To specify the managed nodes reported, use the following syntax:

```
# scrhist –n elsa
```

This command limits the report to only the node elsa.

Checking Node Differences

One of the most useful features of SCR is the ability to report changes to a node over time, or differences between nodes at specified times. The `scrdiff` command reports these differences. Configuration differences are flagged with an asterisk(*) at the beginning of a line.

Time Difference on a Node

If you want to see what changed in node `elsa` between February 14 and April 10, 1999, enter the following command to produce a report that resembles the example below it.

```
# scrdiff elsa:199902141205 elsa:latest
```

This command calls for configuration differences in the node elsa between 12:05 February 14, 1999 and the latest configuration report available. The report looks like the following example:

```
[BASE] elsa : 02/14/1999 12:05 PST

[TARGET] elsa : 04/10/1999 13:05 PDT (latest)

  COMPONENT NAME                          BASE      TARGET
      GROUP NAME
          ATTRIBUTE NAME
  “HP-UX Standard Groups Definition”
      “Host File System”
*         [Host File System Index]        -         11
*         [Mount Point]                   -         lohengrin:/SCR
*         [Remote Mount Point]            -         lohengrin
*         [File System Type]              -         nfs
*         [File System Access]            -         “0:readWrite”
           :
        :
    :
```

The result indicates that between the times specified, a file system, `/SCR`, was mounted on `elsa`.

Difference Between Nodes

The `scrdiff` command can also report the differences between two managed nodes. The following command compares the latest configuration of two nodes, `ortrud` and `elsa`.

```
# scrdiff ortrud:latest elsa:latest
[BASE] ortrud:04/10/1999 13:25 PDT  (latest)
[TARGET] elsa:04/10/1999 13:05 PDT  (latest)
  COMPONENT NAME                    BASE             TARGET
   GROUP NAME
     ATTRIBUTE NAME
  “HP-UX Standard Groups Definition”
   “Host Processor”
    [Host Processor Index]           1                 1
*   [Processor Firmware ID    “HP PA RISC1.1”    “HP PA RISC2.0”
*         [Processor Load]           1                 1

*         [Host Processor Index]     -                 1
*         [Processor Firmware ID]    -           “HP PA RISC2.0”
*         [Processor Load]           -1
           :
```

The result highlights the processor difference. ortrud is a PA 1.1 single processor model, and elsa is a PA 2.0 multi processor model.

Using Tags to Simplify Node Management

As you have already noticed, configuration designations are long. It is tedious to type them over and over, and it can be easy to mistype them. SCR allows you to replace a configuration designation with an easy to use tag. There are two types of tags in SCR, system managed and user defined.

- System managed tags (oldest and latest) are used automatically to indicate the oldest and latest configuration for each node. These tags are managed by the system and updated for you as needed.
- User defined tags are used to represent other system configuration snapshots. If you tag a configuration, SCR assumes it is important to you and does not automatically delete it when it would expire. You can delete the configuration with scrdelete when you want to. You can remove the tag with scrtag.

Use the scrtag command to create or use tags. For example, suppose the node elsa has frequent configuration changes lately, as shown below.

```
# scrhist
NODE        TIME                        ERR         TAG
elsa        02/07/1999 12:05 PST                    oldest
            04/14/1999 13:05 PDT
            04/15/1999 13:05 PDT
            04/17/1999 13:05 PDT
            04/18/1999 13:05 PDT                    latest
```

You can use the oldest and latest tags in future commands, instead of typing the date and time they represent (02/07/1999 12:05 PST and 04/18/1999 13:05 PDT). You can also create tags to use as labels for the other snapshots.

The following command generates the tag, StandardConfig, as a label for the time, 199904171305.:

```
# scrtag –a elsa StandardConfig 199904171305
```

Then, the scrtag command generates a report that identifies current tags, similar to the following example.

```
# scrtag
NODE              TAG                 TIME
elsa              StandardConfig      04/17/1999 13:05 PDT
                  latest              04/18/1999 13:05 PDT
                  oldest              02/07/1999 12:05 PST
```

In the future, you can use the StandardConfig tag instead of typing the date and time in commands. Tags can be modified or deleted using options with scrtag. For example, to move the tag, StandardConfig, for node elsa to the next snapshot (now labeled “latest”), enter the following command:

```
# scrtag –m elsa StandardConfig 199904181305
```

To remove a tag, enter the following command:

```
# scrtag –d elsa StandardConfig
```

This command removes only the tag, not the configuration.

Using Filters

Component information collected by SCR might include information you do not want to see. With `scrfilter`, you can create a view or display filter to limit the information SCR shows you or prints. While the `scrfilter` command generally does not change what information is collected (only limiting the information you see or print), the probe filter (see "Using the Probe Filter" on page 29) actually determines the data collected by SCR.

With the `scrfilter` command, you can designate a filter extension flag for a MIF component, group, or attribute. If the extension flag is `on`, information about the object will be viewed. If it is `off`, the information will not be viewed.

- "Simple Filter Example" on page 26
  - — "Default scrviewer Report" on page 26
  - — "Applying a Filter" on page 27
  - — "Editing a Filter Definition" on page 27
- "Standard Filters" on page 28
- "Using the Probe Filter" on page 29

Simple Filter Example

For example, suppose the “HP-UX Installed Software Definition” component is not interesting to you all the time. The following example shows the information that `scrviewer` reports by default for the HP-UX Installed Software Definition component.

Default scrviewer Report

```
#scrviewer elsa:199804181305

elsa:04/18/1998 13:05 PDT

COMPONENT NAME                VALUE
    GROUP NAME
        ATTRIBUTE NAME
"HP-UX Installed Software Definition"
    Subproducts
        "scr dmi class"       HPUX_Subproducts_
        "scr dmi version"     001
        "scr dmi key"         "Subproduct Software Specification"


   [Subproduct Software Specification]      C-ANSI-C.Development,
                              l=/opt/ansic,r=B.10.20.09,
                              a=HP-UX_B.10.20_700/800,v=HP
        [Create Time]
   [Description]              "Programs that are needed
                              to run C compiler."
        [Modification Time]
        [Size]                8072794
        [Title]               "C Runtime environment"
        [Contents]            C


   [Subproduct Software Specification]      C-ANSI-C.Help,l=
                              /opt/ansic,r=B.10.20.09,
                              a=HP-UX_B.10.20_700/800,v=HP
        [Create Time]
   [Description]              "GUI On_line help utility for C/ANSI
C"
        [Modification Time]
        [Size]                1008074
        [Title]               "On-line Help"
        [Contents]            C-HELP
                              .
                              .
                              .
```

Applying a Filter

To filter the report so this component information will not appear, perform the following steps:

1. Use `scrfilter` to create your own view filter named “NoSoftware”:

   ```
   # scrfilter –c Template NoSoftware
   ```

   This command, makes a copy of the standard filter, Template, and names it `NoSoftware`. Template is one of the standard filters described under the next heading.

2. Use scrfilter to set the filter extension flag to keep the component’s attributes out of the report.

   ```
   # scrfilter –s NoSoftware ”HP-UX Installed Software Definition” off
   ```

   The `-s` option sets the filter for the component to off, as specified after the component name.

3. Use the `NoSoftware` filter with the scrviewer command, as shown below:

   ```
   # scrviewer –f NoSoftware elsa:199904181305
   ```

   The `-f` option tells scrviewer to use the filter that follows it. The HP-UX Installed Software component is excluded from the report. You will see none of the information shown in the example.

Suppose you only wanted to filter out a group from the HP-UX Installed Software Definition component and not the entire component. You could filter the Subproducts group from the report with the following command:

```
# scrfilter –p NoSoftware ”HP-UX Installed Software Definition” \
    “Subproducts” off
```

Since the example showed subproducts information, you would see none of the information shown above. You would, however, see a report about bundles and products and all the other information associated with the HP-UX Installed Software Definition component.

Editing a Filter Definition

You can modify a filter by sending the definition to a file, editing the file and then loading the modified filter definition. This method is recommended for changing more than one value. See “scrfilter (1M)” on page 53 for details.

1. Dump a filter definition to a file by executing:

   ```
   # scrfilter -o filterName fileName
   ```

2. Edit the filter definition now contained in *`fileName.`*

3. Input the modified filter definition by executing:

   ```
   # scrfilter -i fileName
   ```

Standard Filters

SCR comes with nine standard view/display filters that HP has defined for specific components. They are:

- Disk
- FileSystem
- LVM
- Network
- Probe (special, see “Using the Probe Filter” on page 29)
- Software
- SystemProperty
- Patch
- Template (for copying use only)

The Template filter is meant to be used as a template to help you create new view/display filters for your specific needs. You copy it and change the copy as described above. The other standard view/display filters are designed to display the information we suppose you would most likely want about certain DMI groups. Please change them to meet your needs. Changing them will have no effect on system resources.

See “scrfilter (1M)” on page 53 in Chapter 3 for information about creating and modifying view filters.

Using the Probe Filter

The Probe filter determines what information is collected from DMI components, groups, and attributes. The Probe filter determines what is stored in the SCR Repository. You can set Probe filter extension flags to collect (`on`) or not collect (`off`) specific information from the nodes, or to collect or not collect from newly detected components, groups, or attributes.

For a description of the Probe filter, see “scrfilter (1M)” on page 53 in Chapter 3.

We have set the Probe filter to maintain a balance, in usual system environments, between capturing too much information and not capturing enough. We recommend that you play with SCR and tune the Probe filter until it meets your needs. Try various settings before releasing it to the production environment, keeping in mind the following considerations:

- If you capture too much information, the process may take too much time and use a lot of system resources.
- If you capture too little information, you might miss data that is important to managing your system.
- The Probe filter setting is a large factor in determining the number of nodes you can manage from a server. It determines the Repository size and the time required to capture data from each node. A full snapshot on a mid-range server could take more than an hour and produce 8 MBytes of data.
- SCR does not store a snapshot unless something in it has changed since the previous snapshot. Since certain DMI values, such as “Used Disk Space”, change frequently, Probe is set by default not to collect such attributes. If you change it to collect such information, every snapshot might be stored in the Repository.

If you change the Probe filter settings, you can see the effect by taking a new snapshot and running scrdiff on the old and new snapshots. The output of scrdiff indicates missing data with a dash (-).

Additional Uses of SCR

With the above procedures, you have configured SCR to begin collecting and reporting node data. You can further customize SCR's powerful commands using the information available in Chapter 3 and the man pages.

In addition to the procedures described in this chapter, you can use SCR to help produce traditional management information, such as inventory tracking and change management.

- "Inventory Tracking" on page 30
- "Change Management" on page 31
- "Disaster Recovery" on page 32

Inventory Tracking

You can use SCR to generate a system inventory document. You can use this document to maintain inventory, plan for future changes, and help third-party vendors establish what you need. Use the following command to generate such a document:

```
#  scrviewer elsa:latest | lp -dprinter
```

This command generates and prints a complete list of each system's configuration. You could direct the output to a file and use scripts or a word processor to format it into a report.

Be aware that a complete printed list could be very long. You might prefer to print a subset using the view/display filters described in the previous section.

Change Management

SCR can complement your system change management process by keeping good records of changes and alerting you to unplanned changes.

- You can configure SCR to automatically take periodic snapshots.
- You can take snapshots whenever you want them.
- You can tailor reports to compare the current system state to previous states.
- You can generate records of all reports.

Use the `scrupdate`, `scrconfig`, and `scrhist` commands to perform change management, as follows:

1. Complete the initial configuration steps described in this chapter.
2. Take an initial snapshot of the systems as they are, to provide a base for detecting future changes:

   ```
   # scrupdate -a
   ```

3. Set SCR to collect data weekly or as often as you wish:

   ```
   # scrconfig -i lw
   ```

4. Have SCR email a system history report to you. (Remember that SCR will not store a snapshot if the node has not changed since the last snapshot.)

   ```
   # scrhist
   ```

5. When you receive the email, order a scrdiff report on the system nodes that showed differences, to see what they are.

   ```
   # scrdiff node1:time node2:time
   ```

Disaster Recovery

SCR can also help with disaster recovery by showing you a comparison of the system state before and after recovery. To make disaster recovery most effective, set `scrconfig` to collect data weekly or more often, as described in step 3 above.

1. After restoring your system, ensure that DMI is installed and all nodes are configured for access by the CMS.
2. Collect data from the restored system:
   `# scrupdate -n node1`
3. Make sure the data collection was successful:

   `# scrstatus`
4. Compare that data with the data that existed before the disaster recovery:

   `# scrdiff node1:time node1:lastest`
5. Make changes to the system to complete the restoration, if necessary.

This disaster recovery process can be used for more routine situations, such as preparing for a reboot or an operating system upgrade.

As you see, SCR is a powerful and flexible tool for system administration and management. You may devise uses for it that we have not thought of!

SCR Reference (man Pages)

The System Configuration Repository extracts snapshots of a systems configuration by request and/or schedule and stores it as a record in the SCR persistent repository.

The SCR is described in:

- `scr(5)` - System Configuration Repository: page 34.

Ten commands are available for managing data with SCR:

- `scrconfig(1m)` – Configure SCR software: page 43
- `scrdelete(1m)` – Delete specific configuration information: page 48
- `scrdiff(1m)` – Report differences between two sets of configuration information: page 48
- `scrfilter(1m)` – Generate, modify or delete view filter: page 53
- `scrhist(1m)` – Report configuration history: page 64
- `scrstatus(1m)` – Report data collection status table: page 66
- `scrtag(1m)` – Manage tag names for snapshots: page 69
- `scrupdate(1m)` – Collect configuration information: page 72
- `scrviewer(1m)` – Display configuration information of a specified node at a specified time: page 74
- `scrlog_viewer(1m)` – SCR log file viewer: page 77

scr (5)

NAME

SCR – System Configuration Repository

DESCRIPTION

The System Configuration Repository (SCR) extracts a static snapshot of a system's configuration by request and/or by schedule, and stores it as a record in SCR persistent repository. Configuration information is extracted from systems using the Desktop Management Interface (DMI) program routines. Tools are available to identify differences between records stored in the repository. Configuration snapshots will include all relevant configuration information covered by HP-UX System Management Information Format (MIF) and HP Software MIF.

SCR consists of seven components: Configuration Management, Data Management, Filter Management, Tag Management, Presentation, Repository and Log.

Configuration Management commands `scrconfig` maintains the configuration data collection schedule and managed node list.

`scrstatus` generates summary status of data collection activity.

Data Management commands `scrupdate` collects configuration data from managed nodes through DMI, and stores the data in database.

`scrdelete` removes specified or expired configuration data from the database.

`scrupdate` is invoked automatically according to the data collection schedule set by `scrconfig`.

Filter Management command `scrfilter` creates and updates view filter(s) stored in a filter database, which can be used with `scrviewer` or `scrdiff`.

Presentation Component commands `scrhist`, `scrdiff`, and `scrviewer` retrieve stored configuration information and display it in human readable format.

Tag Management command `scrtag` manages tags for user friendly naming of configuration snapshots. Tags can be used as input to the view commands, `scrdiff` and `scrviewer`.

Repository Component provides services for storing configuration snapshots.

Log Component command `scrlog_viewer` displays all SCR commands activities in a graphical user interface (see *samlog_viewer* (1M)).

Commands

SCR can be created, displayed, and manipulated with the following commands:

`scrconfig` configure and query configuration parameters of SCR

`scrdelete` remove SCR configuration information from the database

`scrdiff` list differences between data in SCR configuration snapshots

`scrfilter` list, set, create, modify, and delete SCR view and collection filters

`scrhist` list existing SCR snapshots

`scrstatus` list status of SCR data collection

`scrtag` list, create, modify, and delete SCR tag names

`scrupdate` update SCR contents

`scrviewer` view SCR configuration information

`scrlog_viewer` view and/or save SCR logfile

SCR Logging

Actions taken by SCR are logged into the SCR log file `/var/opt/scr/log/scrlog.log`. The log entries in this file can be viewed with the SCR command `scrlog_viewer`. `scrlog_viewer` can filter the log file by the creation time of the log entry and by the level of detail.

Dependencies Between Software

managed nodes The `Desktop Management Interface for HP-UX` product must be installed on managed nodes.

management server The `Desktop Management Interface for HP-UX` product must be installed and functional before SCR is installed. SCR needs to be installed on the management server.

EXAMPLES

SCR Configuration

- Register the management server on managed nodes.

  The system administrator adds the management server name to the `/var/dmi/dmiMachines` file on each managed node to allow access from management server.

- Register a managed node on the management server.

  Use the `scrconfig` command to maintain managed nodes. All managed nodes must be registered using this command. For example, type in the following line on your management server to include managed node `node2`.

  ```
  # scrconfig -n +node2
  ```

  If the DMI environment of the managed node is not working, not configured correctly, or does not allow access from the management server, the request to register the managed node may fail.

- For scheduled data collection, set the data collection schedule with the `scrconfig` command. For example, type in the following line to set the scheduled data collection time to 12:05 Feb 7, 1999:

```
# scrconfig -n node2 -s 199902071205
```

  In order to confirm current parameters, use the following command:

```
# scrconfig -n node2 -l
```

  The result resembles:

```
NODE    SCHEDULE TIME         INTERVAL  EXPIRATION  TIMEOUT
node2   02/07/1999 12:05 PST   1 day     3 months    5
minutes
```

  Use the `scrconfig` command without options to view all nodes.

- For data collection, the above command is sufficient, no further action is necessary. If scheduled data collection is not needed, make sure scheduled data collection is disabled. The schedule can be stopped with

```
scrconfig -n node2 -s off
```

- Data expiration and data collection timeout parameters can be customized with the `-e` and `-t` options. See the *scrconfig* (1M) manual page for details.

Collecting Configuration Data into SCR

- For scheduled data collection, no further action is required. A daemon process is invoked by the init process and does all work.
- For on-demand data collection, use `scrupdate`. For example, type in the following line to initiate data collection to managed node `node1`:

```
# scrupdate -n node1
```

To check the data collection status, use the `scrstatus` command, which provides a summary report for last, current, and scheduled data collection in time order.

```
# scrstatus
TIME      (START - STOP)      NODE    STATUS     DETAIL
02/06/1999 12:05 - 12:08 PST  node2   Completed
02/06/1999 12:15 - 12:17 PST  node1   Warning     - AcErr -

02/07/1999 16:37 -       PST  node1   Executing

02/07/1999 00:05 -       PST  node1   Scheduled
02/07/1999 12:05 -       PST  node2   Scheduled
```

The first entry above shows past data collection. The second entry shows incomplete data collection, and the detail entry indicates the node was not accessible. The third entry shows data collection currently being performed. The last two entries show data collection scheduled in the future.

Track Configuration Changes with SCR

- List configuration update time for managed nodes with the `scrhist` command. This generates a brief listing of node, time and tag information. This list is the first step in looking at configuration changes. For example, the output of the `scrhist` command may be:

```
# scrhist
NODE             TIME                    ERR  TAG
node1            03/23/1999 12:25 PST         oldest
                 04/10/1999 13:25 PDT         latest
node2            02/07/1999 12:05 PST         oldest
                 02/14/1999 12:05 PST
                 04/10/1999 13:05 PDT         latest
```

All configuration snapshots are listed above.

Administrators can limit the scope of the list by time period and managed nodes, as required. For example, to limit the time period, use the `-p` option:

```
# scrhist -p 19990101 19990226
# scrhist -p 19990401 latest
```

To limit managed nodes, use the `-n` option:

```
# scrhist -n node2
```

- Check modification points with the `scrdiff` command, which lists configuration differences. Configuration differences are indicated with an asterisk(*) at the beginning of each line. For example, the change to managed node, node2, at two time periods is displayed below.

```
# scrdiff node2:199902141205 node2:latest
[BASE] node2 : 02/14/1999 12:05 PST
[TARGET] node2 : 04/10/1999 13:05 PDT (latest)
No filter applied.
  COMPONENT NAME                        BASE            TARGET
      GROUP NAME
          ATTRIBUTE NAME
  "HP-UX Standard Groups Definition"
      "Host File System"
*         [Host File System Index]    -               11
*         [Mount Point]               -
node4:/SCR
*         [Remote Mount Point]        -               node4
*         [File System Type]          -               nfs
*         [File System Access]        -
1:Read-Write
           :
        :
     :
```

The output indicates the change is the mounted file system `/SCR` provided by node4. The difference between different managed nodes can also be displayed. For example, to compare two nodes, node1 and node2:

```
# scrdiff node1:latest node2:latest
[BASE] node1 : 04/10/1999 13:25 PDT (latest)
[TARGET] node2 : 04/10/1999 13:05 PDT (latest)
No filter applied.
  COMPONENT NAME                    BASE             TARGET
      GROUP NAME
          ATTRIBUTE NAME
  "HP-UX Standard Groups Definition"
      "Host Processor"
          [Host Processor Index]    1                 1
*         [Processor Firmware ID]   "HP PA RISC1.1"  "HP
PA RISC2.0"
          [Processor Load]          1                 1

*         [Host Processor Index]    -                 1
*         [Processor Firmware ID]   -                 "HP
PA RISC2.0"
*         [Processor Load]          -                 1
           :
```

The output describes a processor difference: base node1 is a PA 1.1 single processor model, and target node2 is PA2.0 multi processor model.

View configuration information by time and managed node with the scrviewer command, which lists one set of configuration information at a time on a node.

```
# scrviewer node2:latest
node2 : 04/10/1999 13:05 PDT (latest)
No filter applied.
COMPONENT NAME                    VALUE
    GROUP NAME
        ATTRIBUTE NAME
"HP-UX Installed Software Definition"
      :
"HP-UX Standard Groups Definition"
      :
    "General Information"
        [System Name]             node2
        [System Location]         W-206
      :
```

Using Tags and Filters

- Assign easy to recognize labels on configuration information by creating and applying tags. There are two types of tags:

  System managed tags (oldest and latest): These tags indicate the latest and oldest snapshot for each node. These tags are managed by the system and updated whenever data is collected or removed.

  User defined tags: These tags are used to label snapshots. The administrator can define, modify, and remove them.

  For example, suppose the managed node, node2, has recently had frequent configuration changes.

```
# scrhist
NODE            TIME                    ERR  TAG
node2           02/07/1999 12:05 PST         oldest
                04/14/1999 13:05 PDT
                04/15/1999 13:05 PDT
                04/17/1999 13:05 PDT
                04/18/1999 13:05 PDT         latest
```

  A specific time can be labeled with a user defined tag. For example, the following command generates the tag name, StandardConfig:

```
# scrtag -a node2 StandardConfig 199904171305
# scrtag
NODE              TAG                  TIME
node2             StandardConfig       04/17/1999 13:05
PDT
                  latest               04/18/1999 13:05 PDT
                  oldest               02/07/1999 12:05 PST
```

- Tags can be modified or deleted. For example, to move the tag StandardConfig for node node2 to the next (latest) snapshot, use the command:

```
# scrtag -m node2 StandardConfig 199904181305
```

  If a tag is not used, remove it with the scrtag command, for example:

```
# scrtag -d node2 StandardConfig
```

- View filters can be used by administrators to define customized views of SCR information. Any user defined view filter can be applied to any snapshot. For each DMI MIF component, group, or attribute a filter can be set to on (visible) or off (masked). For newly detected MIFs, a filter's extension flag can be set to on (visible) or off (masked).

  For example, suppose the software configuration data is not needed, the HP-UX Installed Software Definition component can be

masked.

```
#scrviewer node2:199804181305
node2 : 04/18/1998 13:05 PDT
No filter applied.
COMPONENT NAME                              VALUE
    GROUP NAME
        ATTRIBUTE NAME
"HP-UX Installed Software Definition"
    "Bundle Contents"
        "scr dmi class"
HPUX_BundleContents_
        "scr dmi version"                   001
        "scr dmi key"                      "Bundle Software
Specification,Index"

        [Bundle Software Specification]
B3899BA,r=B.11.01.01
        [Index]                             001
            :                                :
"HP-UX Standard Groups Definition"
    "DNS Configuration"
        "scr dmi class"
HPUX_BundleContents_
        "scr dmi version"                   001
        "Domain Name"                       dom1.hp.com
            :                                :
```

To mask the component, the following steps can be taken:

1. Create a view filter named `filter1`.
2. `# scrfilter -c Template filter1`
3. Set the mask for the component with
4. `# scrfilter -s filter1 "HP-UX Installed Software Definition" off`
5. Use the `filter1` filter with the `scrviewer` command to successfully exclude the `HP-UX Installed Software Definition` component.
6.
```
# scrviewer -f filter1 node2:199904181305
node2 : 04/18/1998 13:05 PDT
Filter "filter1" applied.
COMPONENT NAME                  VALUE
    GROUP NAME
        ATTRIBUTE NAME
"HP-UX Standard Groups Definition"
    "DNS Configuration"
        "scr dmi class"         HPUX_BundleContents_
```

```
            "scr dmi version"       001
            "Domain Name"           dom1.hp.com
                :                     :
        "General Information"
            "System Name"           node2
                :                     :
```

Similarly, the `DNS Configuration` group of `HP-UX Standard Groups Definition` component can be masked off with the command:

```
# scrfilter -s filter1 "HP-UX Standard Groups Definition"
\\\
"DNS Configuration" off
```

FILES

`/opt/scr/bin` Contains SCR executable files.

`/opt/scr/lbin` Contains SCR backend executables for other commands.

`/opt/scr/lib` Contains SCR shared libraries and SCR tcl extension libraries.

`/opt/scr/share/man` Contains SCR online documentations.

`/var/opt/scr/db` Contains repository, configuration, tag and status files.

`/var/opt/scr/log` Contains commands and backends log file.

`/var/opt/scr/tmp` Temporary directory for using SCR commands and backends.

SEE ALSO

scrconfig (1M), *scrdelete* (1M), *scrdiff* (1M), *scrfilter* (1M), *scrhist* (1M), *scrstatus* (1M), *scrtag* (1M), *scrupdate* (1M), *scrviewer* (1M), *scrlog_viewer* (1M), *dmi* (5).

scrconfig (1M)

NAME

scrconfig – configure and query configuration parameters of SCR

SYNOPSIS

```
/opt/scr/bin/scrconfig [-f outfile]

/opt/scr/bin/scrconfig -l [-n node][-f outfile]

/opt/scr/bin/scrconfig -d[-f outfile]

/opt/scr/bin/scrconfig -n [{+ | - }node ...][-f outfile]

/opt/scr/bin/scrconfig -s [schedule_time | off][-n node][-f
outfile]

/opt/scr/bin/scrconfig -i [interval | off][-n node][-f
outfile]

/opt/scr/bin/scrconfig -e [expiration | off][-n node][-f
outfile]

/opt/scr/bin/scrconfig -t [timeout | off][-n node][-f
outfile]
```

DESCRIPTION

`scrconfig` provides a reference and update interface for System Configuration Repository (SCR) specific configuration parameters. The configuration parameters are:

Managed Node List hostnames targeted by management

Data Collection Schedule next scheduled collection time and current collection interval

Database Expiration amount of time to keep new information

Probe Timeout length of time `scrupdate` will attempt to collect configuration data from a node

The latter three parameters can be customized for each managed node.

Options

`-d` List the default parameter values. The parameters listed are managed node list, probe schedule, data expiration, and timeout.

`-e [`*`expiration`*` | off]` If an expiration period is not specified, list the default expiration period. If the expiration period is specified, the expiration period of the repository is set. The format of *`expiration`* is a numeric value in the

range 1 to 99 with a suffix of y(ear) or m(onth) or the format of *expiration* is a numeric value in the range 1 to 14 with a suffix of w(eek). The initial value is `3m` (three months). If the keyword `off` is specified, old data removal based on the expiration period is disabled. The default expiration period is applied to nodes registered after the expiration period is set.

`-f` *outfile* Send output to specified file *outfile*.

`-i` [*interval* | `off`] If *interval* is not specified, list the default interval. If *interval* is specified, the data collection interval is set. The *interval* is a numeric value in the range 1 to 99 with a suffix of d(ay) or h(our) or the *interval* is a numeric value in the range 1 to 14 with a suffix of w(eek). The initial value is `1w` (one week). If the keyword `off` is specified, rescheduling data collection based on the interval time is disabled. The default interval is applied to nodes registered after the interval is set. If the `-n` option with node name is provided with the `-i` option, list or set the interval time of the specified node.

`-l` List current parameter values. The parameters consist of managed node list, collection schedule, data expiration, and timeout.

`-n` [{+|-}*node* ...] If *node* is not specified, list the managed node list. If *node* is specified, update the managed node list with *node*. *node* with prefix `+`(plus) is added to the list, and with prefix `-`(minus) is removed from the list. On removal, the database contents are also removed by `scrdelete`, which is invoked by `scrconfig`. The node name must be a valid hostname and DMI must be functional for a node to be added.

`-s` [*schedule_time* | `off`] If *schedule_time* is not specified or the keyword `off` is specified, list the next valid default schedule time. If *schedule_time* is specified, the next schedule time is set. If `off` is specified, monitoring activity is disabled. The format of *schedule_time* is *YYYYMMDD*[*hh*[*mm*]]. If *mm* (minute) is not specified, `05` is assumed. If *hh* (hour) is not specified, `00` is assumed. The default schedule time is applied to nodes registered after the schedule time is set. If the `-n` option with node name is provided with the `-s` option, list or set the next scheduled time of the specified node.

`-t [`*`timeout`* `| off]` If *`timeout`* is not specified, list the default data collection timeout value. If *`timeout`* is specified, set the data collection timeout value to the specified value. If the keyword `off` is specified, no timeout will occur. The timeout format is a numeric value in the range 1 to 99 with a suffix of `h`(our) or `m`(inute). The initial value is `15m` (fifteen minutes). The data collection unit is based on the MIF component. If the execution time to collect configuration data exceeds the timeout value, the data collection is canceled. If the `-n` option with node name is provided with the `-s` option, list or set the timeout value of the specified node.

Formats

If configuration parameters or managed nodes are modified, some of the following messages are output.

```
"node" is registered as a
managed node. Default parameters are applied:
    Schedule time:       MM/DD/YYYY hh:mm tz
    Interval:            n days
    Expiration period:   n months
    Collection timeout:  n minutes

"node" is removed from the managed node list.

Default parameter is set to:
    Expiration period:   off

Parameter for
"node" is set to:
    Schedule time:       off
```

EXTERNAL INFLUENCE

Environment Variable

`TZ` TZ determines the conversion between the system time in UTC and the time in the user's local time zone. See *environ* (5) and *tztab* (4).

RETURN VALUE

Upon completion, `scrconfig` returns one of the following exit values:

0	Successful completion.
1	Syntax error.
2	Insufficient privilege.
3	System resource error - process, memory, or files.
4	Runtime error - cannot locate commands or files.
5	Communication error - DMI or network problem.
6	Specified node is not registered.
7	Specified node is already registered.

EXAMPLES

List Managed Nodes and Parameters

View current parameters:

```
    # scrconfig -l
   NODE         SCHEDULE TIME         INTERVAL   EXPIRATION
TIMEOUT
   node1        02/01/1999 15:30 PST   1 day      3 months
5 minutes
    node3            Off               1 day      3 months
10 minutes
```

List Managed Nodes

View the managed node list:

```
    # scrconfig -n
    node1
    node3
```

Update Managed Node Targets

Add `node2` and remove `node1` from managed node list:

```
    # scrconfig -n +node2 -node1
```

Set Data Collection Schedule

Set next scheduled time for `node3` to `06:00 Feb 7th, 1999` in the local time zone:

```
    # scrconfig -n node3 -s 199902070600
```

Set Data Collection Interval Time

Set the interval time for node1 to one week:

```
# scrconfig -n node1 -i 1w
```

Stop Data Collection Schedule

Stop monitoring activity for node2:

```
# scrconfig -n node2 -s off
```

NOTE

Default parameters are effective only for managed nodes registered after the default value is set or modified. The value is not applied to the nodes registered before the default value is modified.

FILES

`/var/opt/scr/log/scrlog.log` Contains managed node addition or deletion, parameter modification log.

SEE ALSO

scrdelete (1M), *scrupdate* (1M), *scr* (5).

scrdelete (1M)

NAME

scrdelete – remove SCR configuration information from the database

SYNOPSIS

`/opt/scr/bin/scrdelete` {`-a` | `-n` *node*} [`-f`] [`-p` *time1* [*time2*]]

DESCRIPTION

`scrdelete` removes specified configuration information from the System Configuration Repository. `scrdelete` is invoked by `scrupdate` to maintain the expiration period. The expiration period is set by `scrconfig`. Without the `-p` option, `scrdelete` removes expired configuration information. With the `-p` option, `scrdelete` removes specified configuration information. Examine configuration information of all nodes with the `-a` option, or of specified nodes with the `-n` option. If configuration information is associated with user defined tags by `scrtag`, the information is not removed without the `-f` option.

Options

`-a` Examine configuration information of all nodes to remove.

`-f` Force removal of configuration information associated with user defined tags, in addition to information not associated with any user defined tags.

`-n` *node* Examine configuration information of the specified node to remove.

`-p` *time1* [*time2*] Remove configuration information within a specific time range or at a specific time. *time1* and *time2* must have the format *YYYYMMDD*[*hh*[*mm*]]. If the hour and minute are not specified, the defaults are `0000` for *time1* and `2359` for *time2*. If the minute is not specified, the defaults are `00` for *time1* and `59` for *time2*.

Formats

Upon completion the following messages are output to indicate removed configuration information.

`Configuration data for "`*node*`:`*time*`" is deleted.`

EXTERNAL INFLUENCE

Environment Variable

TZ	TZ determines the conversion between the system time in UTC and the time in the user's local time zone. See *environ* (5) and *tztab* (4).

RETURN VALUE

Upon completion, `scrdelete` returns with one of the following exit values:

0	Successful completion.
1	Syntax error.
2	Insufficient privilege.
3	Insufficient System resource - process, file, or memory.
4	Runtime error - cannot locate commands or files.
6	Specified node is not registered.

FILES

`/var/opt/scr/log/scrlog.log` Contains data removal log.

SEE ALSO

scrconfig (1M), *scrtag* (1M), *scrupdate* (1M), *scr* (5).

scrdiff (1M)

NAME

scrdiff – list differences between data in SCR configuration snapshots

SYNOPSIS

```
/opt/scr/bin/scrdiff [-a][-f
filter]

base_node:base_time
target_node:target_time
```

DESCRIPTION

`scrdiff` lists configuration snapshot differences between two nodes at two times: *base_node* configuration at *base_time* and *target_node* configuration at *target_time*, or in one node at two times. *base_node* and *target_node* are hostnames. *base_time* and *target_time* are specified by the format *YYYYMMDD*[*hh*[*mm*]], or by tag name (`latest`, `oldest`, or user defined tag name). (See *scrtag* (1M)). If hour and minute are not specified, the default is 2359. If minute is not specified, the default is 59. If a configuration snapshot is not registered at the specified time, the snapshot effective at the time is displayed instead.

Options

`-a` All components, groups, and attributes are listed whether or not differences exist. An asterisk (*) at the beginning of a line indicates that a difference exists between *base_time* and *target_time*. If the information has not been modified, a space character is displayed.

`-f` *filter* Specified filter name is applied to limit the report. See *scrfilter* (1M) for filter customization.

Formats

Output message format is as follows:

[BASE] ***base_node*** **:** ***MM/DD/YYYY hh:mm tz*** **(*tag_name*)**		
[TARGET] ***target_node*** **:** ***MM/DD/YYYY hh:mm tz*** **(*tag_name*)**		
Filter "*filter_name*" applied.		
COMPONENT NAME	BASE	TARGET
GROUP NAME		

[BASE] *base_node* **:** *MM/DD/YYYY hh:mm tz* **(***tag_name***)**		
[TARGET] *target_node* **:** *MM/DD/YYYY hh:mm tz* **(***tag_name***)**		
ATTRIBUTE NAME		
m component_name	*base_value*	*target_value*
m group_name	*base_value*	*target_value*
m attribute_name	*base_value*	*target_value*

If only attributes differ, the *component_name*, and *group_name* of the attributes are printed out without BASE or TARGET fields. In case of table group, *attribute_name* is displayed with bracket([]), and a null line is inserted between rows. If view filter is not specified, "No filter applied." is displayed. The mark field, *m*, highlights differences between BASE and TARGET by an asterisk(*). If there is no difference, a space is used instead.

EXTERNAL INFLUENCE

Environment Variable

TZ — TZ determines the conversion between the system time in UTC and the time in the user's local time zone. See *environ* (5) and *tztab* (4).

RETURN VALUE

Upon completion, `scrdiff` returns with one of the following exit values:

0	Successful completion.
1	Syntax error.
2	Insufficient privilege.
3	Insufficient System resource - process, file, or memory.
4	Runtime error - cannot locate commands or files.
6	Specified node is not registered.
7	Specified snapshot is not registered in repository.
8	Specified filter is not registered in repository.
9	Specified tag is not registered in repository.

NOTE

Precision of time depends on configuration data collection made by scrupdate.

EXAMPLES

List Two Configuration Difference

List differences between node node1 at 20:52 April 22 1999 and latest configuration of node2.

```
# scrdiff node1:199904231200 node2:latest
[BASE] node1 : 04/22/1999 20:52 PDT
[TARGET] node2 : 04/23/1999 10:15 PDT (latest)
No filter applied.
  COMPONENT NAME                          BASE          TARGET
    GROUP NAME
      ATTRIBUTE NAME
  "HP-UX Standard Groups Definition"
    "General Information"
*     "System Name"                       node1         node2
*     "System Location"                   E-110         W-206
         :                                  :             :
```

FILES

/var/opt/scr/log/scrlog.log Contains execution log.

SEE ALSO

scrfilter (1M), *scrtag* (1M), *scrupdate* (1M), *scr* (5).

scrfilter (1M)

NAME

scrfilter – list, set, create, modify, and delete SCR view and collection filters

SYNOPSIS

```
/opt/scr/bin/scrfilter [ -l [filter [component [group
[attribute ]]]]]

/opt/scr/bin/scrfilter -s filter [component [group
[attribute]]] { on|off}

/opt/scr/bin/scrfilter -S filter
component group attribute {pattern|reset}

/opt/scr/bin/scrfilter -c filter new_filter

/opt/scr/bin/scrfilter -m filter new_filter

/opt/scr/bin/scrfilter -d filter

/opt/scr/bin/scrfilter -u filter

/opt/scr/bin/scrfilter -i file

/opt/scr/bin/scrfilter -o filter file
```

DESCRIPTION

`scrfilter` provides functions to create and control view and collection filter. View filters can be used by `scrdiff` and `scrviewer` to filter out uninteresting configuration data. The collection filter, `Probe`, is used by `scrupdate` to filter out collection configuration data.

`scrfilter` lists defined filter names, creates, deletes and renames filters, displays filter settings, sets view or collection flags for specified component/group/attribute values, and updates filters with current DMI information provided by DMI from the managed nodes.

Filter names must start with an alphabetic character. Optional following characters need to be one of alphabet(A-Z,a-z), digit(0-9), plus(+), hyphen(-) or underscore(_).

The view extension flags, `on` and `off`, are set by copying and modifying the `Template` view filter. `Template` can also be used as a default filter. The collection extension flags, `on` and `off`, detemine whether `scrupdate` collects configuration data for a specified or newly detected component, group, or attribute. See the `-s` and `-u` options for details. See also "List Filter Contents" in the Examples section.

Options

`-c` *`filter new_filter`* Copy *`filter`* to *`new_filter`*.

`-d` *`filter`* Delete *`filter`*.

`-i` *`file`* Modify existing filter by specified filter *`file`*. The *`file`* can be generated by `-o` option. A filter name is described in the *`file`* and the filter needs to exist in repository. If the *`file`* contains new component, group, or attribute, the attempt to modify the filter causes error.

`-l` [*`filter`* [*`component`* [*`group`* [*`attribute`*]]]] List registered filter names or settings specified by *`filter`*. If no argument is specified, all registered filter names are listed. If only *`filter`* is specified, all filter settings of the filter are listed. If further arguments are specified, the setting is listed.

`-m` *`filter new_filter`* Rename *`filter`* to *`new_filter`*.

`-o` *`filter file`* Generate filter *`file`* from specified *`filter`*. The *`file`* describes the filter name and the filter settings. It can be modified by text editor such as `vi`, and be reflected back to the *`filter`* by `-i` option.

`-s` *`filter`* [*`component`* [*`group`* [*`attribute`*]]] {`on`|`off`} Set view or collection flag for specified *`component`*, *`group`*, or *`attribute`* configuration data or set the default for newly detected information. If *`component`*, *`group`*, and *`attribute`* are omitted, the extension flag to control defaults for newly detected information is set. If *`component`*, *`group`*, or *`attribute`* is specified, the view or collection flag for view filters or the collection filter is set.

If `on` is specified for view filters, `scrviewer` and `scrdiff` display the data. If `off` is specified, the data is not displayed by these commands. If `on` is specified for the collection filter, `scrupdate` collects the data. If `off` is specified, the data is not collected by the command. If a sub-parameter, such as *`group`* or *`attribute`*, is set to `on`, and a higher parameter, such as *`component`*, is set to `off` from `on`, no data under the component is displayed or collected.

`-S` *`filter component group attribute`* {*`pattern`*|`reset`} Set or reset search condition to the *`group`* or rows of the *`group`* for specified view *`filter`*. If *`pattern`* is specified for the *`attribute`*, `scrviewer` and `scrdiff` display the *`group`* or the rows of the *`group`* only if the *`attribute`* matches the *`pattern`*. If `reset` is specified for the *`attribute`*,

search condition is reset.

The search condition affects the specified *group* only. Whether other groups are displayed or not depends on each view flag value of groups.

The *pattern* needs to be enclosed by round brackets. Asterisk (*) can be used as wildcard character. Backslash (\) is required as escape character to use asterisk or round brackets as part of the *pattern*. The *pattern* is allowed to exist only one *attribute* per each *group*. The attempt to set second *pattern* into different *attribute* within one *group* is not allowed. The option is prohibited for the collection filter Probe.

`-u` *filter* Update specified filter settings by accessing the configuration data name available on managed nodes. If new data is detected, the data name is imported to the filter. The update option does not remove existing data names. The view or collection flag value for the new data is defined by the extension flag with the exception that `off` is set merely on the highest data. (See NOTE for more information.)

Formats

The filter name list generated by the `scrfilter` command without options or with the `-l` option has the following format:

```
filter_name1
filter_name2
     :
```

The filter settings list generated with the `-l` option specified has the following format:

[Filter Name]: *filter*		
[Data Version]: *data_version*		
[Extension]: *flag*		
Component: *component1*	*flag*	*option*
Group: *group1*	*flag*	*option*
Attribute: *attribute1*	*flag*	*option*

Attribute: *attribute2*	*flag*	*option*
:		

Where *flag* above is either on or off.

Where *option* above can be NewlyDetected, if the *component*, *group*, or *attribute* is detected using -u option. If a *pattern* is set to an *attribute* in repository, scrfilter prints the *pattern* in the *option* field for the *attribute* and keyword Conditional in the same field for the *group*.

The filter file generated with the -o option specified has the following format:

[Filter File Version]: *file_version*		
[Filter Name]: *filter*		
[Data Version]: *data_version*		
[Extension]: *flag*		
Component: *component1*	*flag*	
Group: *group1*	*flag*	
Attribute: *attribute11*	*flag*	*pattern*
Attribute: *attribute12*	*flag*	
Group: *group2*	*flag*	
Attribute: *attribute21*	*flag*	
Attribute: *attribute22*	*flag*	*pattern*
:		

Where *flag* above is either on or off. The syntax of *pattern* is the same as that of *pattern* of search condition (-S option). The only one *pattern* can be specified to a *group*. If two or more *patterns* are specified to a *group*, scrfilter returns with error. Each *group* needs

to be followed by a blank line. The *flag* and *pattern* fields can be modified to customize a *filter* using `-i` option.

`scrfilter` outputs the following messages.

```
Filter
 "original_name"
copied to
"name".

Filter
 "name"
removed.

Filter
 "name"
renamed
 "new_name".

Filter
 "name"
set
 "on/off"
for the following information:

  Component:
component1
  Group:
group1
  Attribute:
attribute2

Filter
 "name"
updated.

No new information detected. Filter
 "name"
unchanged.

Search condition
 "pattern"
for the filter
 "name"
is set to the following information:
```

```
  Component:
component1
  Group:
group1
  Attribute:
attribute2

Search condition
for the filter
 "name"
is reset to
the following information:

  Component:
component1
  Group:
group1
  Attribute:
attribute2

File
 "file"
applied to filter
 "name".
```

RETURN VALUE

Upon completion, `scrfilter` returns with one of the following exit values:

0	Successful completion.
1	Syntax error or update request to overwrite existing filter.
2	Insufficient privilege.
3	Insufficient System resource - process, file, or memory.
4	Runtime error - cannot locate commands or files.
5	Communication error - DMI or network problem.
6	Specified filter is not registered.
7	Specified component, group or attribute is not identified.
8	Collection or template view filter cannot be deleted or moved to rename.
9	Search condition of the group is already set.
10	Specified filter file is incorrect format or data.

11	Filter data version is unexpected.

NOTE

System Configuration Repository provides nine default filters, the collection filter, the template view filter and seven other pre-defined view filters. User-defined view filters can be created by copying the template or another view filter and modifying it. Filter names must be alphanumeric without white space or tabulation.

The filter names `Probe` and `Template` are reserved. Operations to remove or rename both filters are prohibited. The seven other pre-defined view filters can be removed or renamed like user-defined view filters.

`Probe`: Collection filter used by `scrupdate` command. Customers are recommended not to modify these settings. The collection filter disables by default "dynamic information", such as system date, number of processes in DMTF UNIX standard MIF. Enabling these attributes greatly increases the disk space consumed by SCR. This filter is prohibited from removing and renaming by `-d` and `-m` options.

`Template`: Read only template view filter. The filter can be used to generate new filters. It is prohibited to modify the filter by `-d`, `-i`, `-m`, `-o`, `-s`, `-S`, and `-u` options.

`Disk`, `FileSystem`, `LVM`, `Network`, `Patch`, `Software` and `SystemProperty`: These are examples of view filter. These are free to modify, rename and remove like user-defined view filters.

DMI defines hierarchical data levels: component, group, and attribute (in the order of high to low level). If view or collection flags are set different among these levels, the value `off` for higher data has precedence over the value `on` for lower data to display or collect the lower data. For example, if the component is set `off` and group is set `on`, the group is not displayed or collected.

Rules are:

- If all view or collection flags for target and higher data are `on`, then target data is displayed or collected.
- If some view or collection flags for target or higher data are `off`, the target data is not displayed or collected.

View or collection flag for new data detected by update (-u) option is set by the following rules:

- If extension flag is on, all flags for new data are set on.
- If extension flag is off, highest data within the new data is set off, and the rest are set on.

EXAMPLES

List Registered Filters

List all registered filters.

```
# scrfilter -l
Disk
FileSystem
LVM
Network
Patch
Probe
Software
SystemProperty
Template
myFilter
```

List Filter Contents

List filter Template settings.

```
# scrfilter -l Template
[Filter Name]:  Template
[Data Version]: 1
[Extension]:    on
Component:      DMI 2.0 Service Provider        off
  Group:        ComponentID                     on
    Attribute:  Manufacturer                    on
    Attribute:  Product                         on
     :
```

Create Filter

Create filter myFilter from filter Template.

```
# scrfilter -c Template myFilter
```

Set or Reset Extension Flag

Disable automatic extension for newly detected DMI MIF information. Newly detected information is not displayed.

```
# scrfilter -s myFilter off
```

Enable automatic display extension for newly detected DMI MIF information. Newly detected information is displayed by this filter.

```
# scrfilter -s myFilter on
```

Set or Reset Filter for Specific Information

Exclude HP-UX Installed Software Definition component, and include HP-UX Standard Groups Definition component for filter myFilter.

```
# scrfilter -s myFilter "HP-UX Installed Software
Definition" off
# scrfilter -s myFilter "HP-UX Standard Groups Definition"
on
```

List Filter Contents

Confirm current filter parameter for Host Physical Memory group in HP-UX Standard Groups Definition component.

```
# scrfilter -l myFilter "HP-UX Standard Groups Definition"
\\\
"Host Physical Memory"
on
```

Set or Reset search condition

Set search condition for Host Device group. Only the rows in which the Device Type value matches with DiskStorage will be shown by scrviewer.

```
# scrfilter -S DiskDevice "HP-UX Standard Groups
Definition" \\\
"Host Device" "Device Type" "(DiskStorage)"
```

Confirm the search condition of Disk Device filter.

```
# scrfilter -l DiskDevice
[Filter Name]:  DiskDevice
[Data Version]: 1
[Extension]:    off
     :
Component:      "HP-UX Standard Groups Definition"  on
  Group:        ComponentID                         off
     :
  Group:        "Host Device"                       on
Conditional
    Attribute:  "Host Device Index"                 on
    Attribute:  "Device Type"                       on
(DiskStorage)
     :
```

Display the configuration of node node1 applying filter DiskDevice.

```
# scrviewer -f DiskDevice node1:Stable
Filter "DiskDevice" applied.
COMPONENT NAME                          VALUE
    GROUP NAME
        ATTRIBUTE NAME
"HP-UX Standard Groups Definition"
    "Host Device"
    "scr dmi class"                     "HPUX_Host Device_"
    "scr dmi version"                   001
    "scr dmi key"                       "Host Device Index"

       [Host Device Index]              4
       [Device Type]                    6:DiskStorage
       [Device Description]             "HP      C3325A"
      [Device ID]                     "0 0 2 2 0 0 0 0 186 168"
       [Device Status]                  2:Running
       [Device Errors]                  0

       [Host Device Index]              12
       [Device Type]                    6:DiskStorage
       [Device Description]             "HP      C2247"
      [Device ID]                      "0 0 2 2 0 0 0 0 190 67"
       [Device Status]                  2:Running
       [Device Errors]                  0
            :
```

Reset search condition.

```
# scrfilter -S DiskDevice "HP-UX Standard Groups
Definition" \\\
"Host Device" "Device Type" reset
```

Modify filter setting using filter file

Generate filter file `SettingFile` from filter `myFilter`.

```
# scrfilter -o myFilter SettingFile
```

Edit view flags of groups in filter file `SettingFile` using `vi`. The new setting for filter `myFilter` is prepared as follows:

```
[Filter File Version]:  1
[Filter Name]:  myFilter
[Data Version]: 1
[Extension]:    on
     :
Component:      "HP-UX Standard Groups Definition"  on
  Group:        ComponentID                         off
     :
  Group:        "System Contact Information"        off
     :
```

Apply the new setting from filter file `SettingFile` to filter `myFilter`.

```
# scrfilter -i SettingFile
```

FILES

`/var/opt/scr/log/scrlog.log` Contains filter creation, deletion, and modification log.

SEE ALSO

scrdiff (1M), *scrupdate* (1M), *scrviewer* (1M), *scr* (5).

scrhist (1M)

NAME

scrhist – list existing SCR snapshots

SYNOPSIS

```
/opt/scr/bin/scrhist [-n node ...
][-p
start_time  end_time]
```

DESCRIPTION

`scrhist` lists snapshots for target nodes in a given period of time. Target nodes can be specified by `-n` option and period of time is specified by `-p` option.

Options

`-n` *node* Specify target node name(s). If not specified, all managed nodes are targeted.

`-p` *start_time end_time* Specify period of time by *start_time* and *end_time*. The format is *YYYYMMDD*[*hh*[*mm*]]. *start_time* and *end_time* can be any tag, including the keywords `oldest` and `latest`. If hour and minute are omitted, the defaults are `0000` for *start_time*, and 2359 for *end_time*. If minute is omitted, the defaults are `00` for the *start_time*, and 59 for the *end_time*.

Formats

The output to standard out has the following format:

NODE	TIME	ERR	TAG
node	*MM/DD/YYYY hh:mm tz*	*error*	*tag*

node is the hostname of the managed node. If the same node appears multiple times, only the first line indicates the node name, the rest of the lines are left blank. The time field format above is *MM/DD/YYYY hh:mm tz*. For normal data collection, the error field is blank. An entry in the error field indicates data collection encountered an error or warning. *error* may take the values: `E` (error) indicates no information is collected. `W` (warning) indicates partial error is reported. *tag* is displayed if tags are associated with the snapshot. See *scrtag* (1M).

EXTERNAL INFLUENCE

Environment Variable

`TZ` TZ determines the conversion between the system time in UTC and the time in the user's local time zone. See *environ* (5) and *tztab* (4).

RETURN VALUE

Upon completion, `scrhist` returns with one of the following exit values:

`0` Successful completion.

`1` Syntax error.

`2` Insufficient privilege.

`3` Insufficient system resource - process, file, or memory.

`4` Runtime error - cannot locate commands or files.

`6` Specified node is not registered.

EXAMPLES

```
# scrhist -n node1 node2 -p 19980101 latest
NODE                TIME                    ERR  TAG
node1               03/23/1998 10:10 PST
                    04/21/1998 09:18 PDT  W
                   06/21/1998 19:56 PDT       StableSystem
latest
node2               01/31/1998 11:23 PST
                    04/21/1998 09:18 PDT
                  06/23/1998 19:48 PDT      Testing latest
```

FILES

`/var/opt/scr/log/scrlog.log` Contains execution log.

SEE ALSO

scrtag (1M), *scr* (5).

scrstatus (1M)

NAME

scrstatus – list status of SCR data collection

SYNOPSIS

```
/opt/scr/bin/scrstatus

/opt/scr/bin/scrstatus -a[-n node]

/opt/scr/bin/scrstatus -r[-n node]
```

DESCRIPTION

`scrstatus` generates a status report of configuration data collection by `scrupdate`. The report shows the last previous collection, any current collection, and the next scheduled collection for currently managed nodes. Previous collection information appears first, with current collection below it, and future scheduled collection below that.

For each node, `scrupdate` shows the time of the last, current, or scheduled collection, the node name, the collection status, and any relevant details.

Options

`-a`	Output detailed statistics information.
`-n` *node*	Limit the report to specified *node* only.
`-r`	Reset status information including last execution status.

Formats

`scrstatus` has two formats. Without `-a` option, the following items are reported.

TIME	(START - STOP)	NODE	STATUS	DETAIL
MM/DD/YYYY	*hh:mm - hh:mm tz*	*node_name*	*status*	*detail*

With `-a` option, the following items are reported.

TIME	(START - STOP)	NODE	STATUS	DETAIL		
	(MEAN MAX)			NML	WRN	ERR
MM/DD/YYYY	*hh:mm - hh:mm tz*	*node_name*	*status*	*detail*		
	hh:mm hh:mm			*n*	*n*	*n*

status is either Completed (without error), Warning (Partial error reported), Error (Full error), Executing (now executing), or Scheduled (in future).

detail is displayed only if *status* field is either Warning or Error. The fifth column, *detail* indicates encountered errors with three values: Tmout (Timeout), AcErr (Access error), and RtErr (Runtime error). If a timeout is detected, the data collection timeout value can be increased with scrconfig. If an access error occurs, either the DMI is not functional or a network error prevents SCR from collecting data. Fix the DMI or network problem. If a runtime error occurs, a resource shortage such as file system, memory, or process resource prevented data collection. Increase the limiting resource.

The MEAN collection time is calculated using Normal (NML) terminations. The MAX (maximum) time is the longest data collection time with Normal (NML - no error) or Warning (WRN - partial error) termination.

n is the total number of NML (Normal), WRN (Warning), and ERR (Error) results.

EXTERNAL INFLUENCE

Environment Variable

TZ — TZ determines the conversion between the system time in UTC and the time in the user's local time zone. See *environ* (5) and *tztab* (4).

RETURN VALUE

Upon completion, `scrstatus` returns with one of the following exit values:

`0`	Successful completion.
`1`	Syntax error.
`2`	Insufficient privilege.
`3`	Insufficient System resource - process, file, or memory.
`4`	Runtime error - cannot locate commands or files.

EXAMPLES

List Status Summary

List status information

```
# scrstatus
TIME      (START - STOP)      NODE       STATUS   DETAIL
04/10/1998 00:05 - 00:09 PDT  node2      Completed
04/10/1998 00:20 - 00:23 PDT  node1      Warning    -
AcErr   -

04/11/1998 00:05 -       PDT  node2      Executing

04/11/1998 00:20 -       PDT  node1      Scheduled
04/12/1998 00:05 -       PDT  node2      Scheduled
```

FILES

`/var/opt/scr/log/scrlog.log` Contains reset request log.

SEE ALSO

scrconfig (1M), *scrupdate* (1M), *scr* (5).

scrtag (1M)

NAME

scrtag – list, create, modify, and delete SCR tag names

SYNOPSIS

```
/opt/scr/bin/scrtag

/opt/scr/bin/scrtag -l node tag

/opt/scr/bin/scrtag -r node time

/opt/scr/bin/scrtag -a node tag time

/opt/scr/bin/scrtag -m node tag time

/opt/scr/bin/scrtag -d node tag
```

DESCRIPTION

`scrtag` provides reference and management interface of SCR tags. The tag is an alias for a configuration snapshot. `scrtag` lists all registered tags, converts snapshot time to and from tag, registers a tag associated with a snapshot, modified the association between tag and time, removes a specified tag.

The tag name must start with an alphabetic character. Optional following characters can be alphabet(A-Z,a-z), digit(0-9), plus(+), hyphen(-) or underscore(_).

Options

`-l` *node tag* Specified *tag* for *node* is converted to time.

`-r` *node time* Specified *time* for *node* is converted to tag.

`-a` *node tag time* Add *tag* for *node* associated with *time*.

`-m` *node tag time* Modify association of the *tag* to *time*.

`-d` *node tag* Delete specified *tag* for the *node*.

Formats

Without option, the `scrtag` output message has the following format:

NODE	TAG	TIME
node_name	*tag_name*	*MM/DD/YYYY hh:mm tz*

With the `-l` or `-r` option, `scrtag` outputs converted time or tag:

YYYYMMDDhhmm
tag_name

With the `-a`, `-m` or `-d` option, `scrtag` outputs the following messages on successful termination:

`"node_name:tag_name" registered for MM/DD/YYYY hh:mm tz.`
`"node_name:tag_name" modified to MM/DD/YYYY hh:mm tz.`
`(former time: MM/DD/YYYY hh:mm tz)`
`"node_name:tag_name" for MM/DD/YYYY hh:mm tz removed.`

EXTERNAL INFLUENCE

Environment Variable

`TZ` — `TZ` determines the conversion between the system time in UTC and the time in the user's local time zone. See *environ* (5) and *tztab* (4).

RETURN VALUE

Upon completion, `scrtag` returns with one of the following exit values:

`0` Successful completion.

`1` Syntax error or update request to system managed tags.

`2` Insufficient privilege.

`3` Insufficient System resource - process, file, or memory.

`4` Runtime error - cannot locate commands or files.

`6` Specified node is not registered.

`7` Specified tag is not registered.

`8` Specified tag is already registered.

`9` Specified snapshot is not registered in repository.

EXAMPLES

List All Tags

List all registered tags

```
# scrtag
NODE                TAG             TIME
node1               Stable          01/31/1999 00:25 PST
                    latest          04/10/1999 01:25 PDT
                    oldest          01/31/1999 00:25 PST
node2               latest          04/10/1999 16:00 PDT
                    oldest          02/07/1999 00:25 PST
```

Display Time Associated with Tag

Display the time associated with tag `Stable` for node `node1`.

```
# scrtag -l node1 Stable
199901310025
```

Set tag

Set tag `Standard` for node `node1` to Feb 7 13:00, 1999:

```
# scrtag -a node1 Standard 199902071300
```

Remove Tag

Remove the tag `Standard` for node `node1`:

```
# scrtag -d node1 Standard
```

NOTE

The System Configuration Repository maintains the latest and oldest tags for each node as `scrupdate` probes each managed node. These tags are controlled by the system and treated as read-only tags. Any attempt to add, modify, or remove these tags are checked out. The expiration period of configuration data can be set so that as time passes, snapshots not associated with user defined tags are removed. Use `scrdelete` with the `-f` option to remove snapshots with associated user defined tags.

FILES

`/var/opt/scr/log/scrlog.log` Contains tag addition, modification and deletion log.

SEE ALSO

scrdelete (1M), *scrupdate* (1M), *scr* (5).

scrupdate (1M)

NAME

scrupdate – update SCR contents

SYNOPSIS

```
/opt/scr/bin/scrupdate [-t name
] {-a | -n node}
```

DESCRIPTION

`scrupdate` collects system configuration information on registered managed nodes through DMI. `scrupdate` compares newly collected information with the most recently collected information. If a difference is detected, newly collected information is registered in the repository. If no difference exists, the new information is not registered in the repository.

Options

`-a`	Collect configuration information from all managed nodes.
`-n` *node*	Collect configuration information for a specified node *node*.
`-t` *name*	Set *name* as a tag for the collected snapshot. `scrdiff` or `scrviewer` can use the tag. See *scrtag* (1M) for more information.

Formats

Upon completion, some of the following messages are output:

`Configuration data for "`*node*`:`*time*`" collected.`

`Tag "`*name*`" generated for "`*node*`:`*time*`".`

```
Configuration data for "node:time"
 collected with partial error.
(Reason_code)
```

```
Cannot collect configuration data for "node:time".
(Reason_code)
```

`Data registered in the repository.`

`No data registered. Same as "`*node*`:`*time*`".`

Reason_code may be one or a combination of `Timeout`, `AccessError`, or `RuntimeError`.

RETURN VALUE

Upon completion, `scrupdate` returns with one of the following exit values:

`0`	Successful completion with or without partial failure.
`1`	Syntax error.
`2`	Insufficient privilege.
`3`	Insufficient System resource - process, file, or memory.
`4`	Runtime error - cannot locate commands or files.
`5`	Communication error - DMI or network problem.
`6`	Specified node is not registered.

EXAMPLES

Collect configuration information for node `node1`.

```
# scrupdate -n node1
Configuration data for "node1:199904101205" collected.
Data registered in the repository.
```

Collect configuration information for node `node2`, and label it with a tag called `Stable`.

```
# scrupdate -t Stable -n node2
Configuration data for "node2:199904101208" collected.
Data registered in the repository.
Tag "Stable" generated for "node2:199904101208".
```

FILES

`/var/opt/scr/log/scrlog.log` Contains data collection log.

SEE ALSO

scrdiff (1M), *scrtag* (1M), *scrviewer* (1M), *scr* (5).

scrviewer (1M)

NAME

scrviewer – view SCR configuration information

SYNOPSIS

/opt/scr/bin/scrviewer [-f *filter*] *node:time*

DESCRIPTION

scrviewer displays configuration information on the specified *node* at the specified *time*. *node* is the hostname of the desired node. The *time* format is *YYYYMMDD*[*hh*[*mm*]], or the keyword oldest, latest, or a user defined tag name registered through scrtag. If hour and minute are omitted, the default is 2359. If minute is omitted, the default is 59. If a snapshot of the specified time is not in the repository, the snapshot effective at the time is displayed.

Options

-f *filter* — Specify view filter name. If not specified, all stored configuration information is reported. See *scrfilter* (1M) for filter management.

Formats

The output message format is as follows:

***node_name*:*MM/DD/YYYY hh:mm tz* (tag_name)**	
Filter "filter_name" applied.	
COMPONENT NAME	**VALUE**
GROUP NAME	
ATTRIBUTE NAME	
component_name	
group_name	
attribute_name	*value*

When a group consists of multiple rows of information, *attribute_name* is displayed with brackets ([]) and a null line is inserted between rows. If view filter is not specified, "No filter applied." is displayed.

EXTERNAL INFLUENCE

Environment Variable

TZ	TZ determines the conversion between the system time in UTC and the time in the user's local time zone. See *environ* (5) and *tztab* (4).

RETURN VALUE

Upon completion, scrviewer returns with one of the following exit values:

0	Successful completion.
1	Syntax error.
2	Insufficient privilege.
3	Insufficient System resource - process, file, or memory.
4	Runtime error - cannot locate commands or files.
6	Specified managed node is not registered.
7	Specified snapshot is not registered in repository.
8	Specified filter is not registered in repository.
9	Specified tag is not registered in repository.

EXAMPLES

List Configuration Information

List the configuration information of node node1 on June 23 12:00 1998 PDT.

```
# scrviewer node1:199806231200
node1 : 06/23/1998 12:00 PDT (Stable)
No filter applied.
COMPONENT NAME                  VALUE
  GROUP NAME
    ATTRIBUTE NAME
"HP-UX Standard Groups Definition"
  "DNS Configuration"
    "Domain Name"               dom1.hp.com
    Search
    "Server IP Address(es)"     192.168.10.10
         :                          :
```

FILES

`/var/opt/scr/log/scrlog.log` Contains execution log.

SEE ALSO

scrfilter (1M), *scrtag* (1M), *scr* (5).

scrlog_viewer (1M)

NAME

scrlog_viewer – view and/or save SCR logfile

SYNOPSIS

`/opt/scr/bin/scrlog_viewer [-s MMDDhhmm[[CC]YY]] [-e MMDDhhmm[[CC]YY]] [-l detail] [-c component] [-u user] [-o ofile] [-t] [-n] [file]`

DESCRIPTION

The `scrlog_viewer` command enables the viewing of part or all of the SCR logfile (or another file containing data in the same format) at varying levels of detail.

The `scrlog_viewer` command executes in either interactive or non-interactive mode, depending on the options given. In non-interactive mode, `scrlog_viewer` filters the log file and writes the resulting data either to stdout or to a destination file, if specified. In interactive mode, `scrlog_viewer` displays a graphical user interface that enables filtering, saving one or more versions of the log file to other files, scrolling back and forth among the logfile entries, etc.

`scrlog_viewer` does not modify the contents of the SCR logfile. The contents of the log file are filtered and displayed according to the settings of the available filters. Multiple instances of `scrlog_viewer` can be run simultaneously.

Filters

`scrlog_viewer` supports four types of filters: level of detail, component, date/time, and user filters. These filters can be used in combination to provide highly selective logfile viewing.

The level of detail filters control how much detail is displayed. The SCR logfile may contain entries of many different types. The entry types currently supported are: summary, detail, error, and note. The level of detail filters display some or all of these entry types, depending on which filter is chosen. The level of detail filters are:

`Summary` Displays only the higher level messages. These include *`summary`*, *`error`*, and *`note`* entry types.

`Detail` Includes `Summary` level of detail, and adds *`detail`* log entries. If no level of detail is specified this is the default.

`Commands Only` Displays only the literal commands that were executed. These commands may include HP-UX commands as well as SCR commands.

The component filters control the component to be displayed. The component filters are:

`Configuration` *scrconfig* (1M), *scrstatus* (1M)

`Data Management` *scrupdate* (1M), *scrdelete* (1M)

`Filter Management` *scrfilter* (1M)

`Tag Management` *scrtag* (1M)

`Presentation` *scrhist* (1M), *scrdiff* (1M) and *scrviewer* (1M)

`Repository` Repository activities

`Other` Reserved for future commands

`All` All of the above components

The date/time filters are used to ask for entries written since a specific date/time, before a specific date/time, or both.

The user filters are supported for compatibility with samlog and dmilog only. SCR requires root privilege to operate.

Options

The following options enable you to set up filtering and other attributes. If `scrlog_viewer` runs interactively, these attributes may also be set and modified in the various supported menus and displays. The available options are:

`-s` *MMDDhhmm [[CC] YY]* The `-s` option sets the start date/time filter to the date/time given by its argument. The date/time is specified in the following format:

MM	Month specified as a two digit number (`01-12`).
DD	Day specified as a two digit number (`01-31`).
hh	Hour specified as a two digit number (`00-23`).
mm	Minute specified as a two digit number (`00-59`).
CC	The first two digits of the desired year (`19-20`).
YY	The last two digits of the desired year (`70-99`, `00-37`).
	The range of valid years without the first two digits (*CC*) is `1970-1999`, `2000-2037`. If the year is specified as a four digit number, the valid range is

1902-2037. If the year is not specified, the current year is used.

If no start time is given, the beginning of the log is used as the start time.

`-e` *`MMDDhhmm [[ CC ]  YY ]`* The `-e` option sets the end date/time filter to the date/time given by its argument. The date/time is specified as described for the `-s` option. If no end time is given, then an end date/time of infinity (no end time) is used.

`-l` *`detail`* The `-l` option sets the desired level of detail. *`detail`* must be specified and may have the value `S`, `D`, `V`, or `C` The level of detail is set as follows:

`S` = `Summary`
`D` = `Detail`
`V` = `Verbose` (Compatibility only)
`C` = `Commands Only`

If the `-l` option is not specified, the default value is `D`.

`-c` *`component`* The `-c` option sets the desired component. *`component`* must be specified and may have the value `C`, `D`, `F`, `T`, `R`, `P`, `O`, or `A`. The component is set as follows:

`C` = `Configuration`
`D` = `Data Management`
`F` = `Filter Management`
`T` = `Tag Management`
`R` = `Repository`
`P` = `Presentation`
`O` = `Other`(reserved)
`A` = `All`

If the `-c` option is not specified, the default value is `A`.

`-u` *`user`* The `-u` option sets the user filter to the user name or user ID specified by *`user`*. Only entries logged by this user are displayed. The option is provided for compatibility with *samlog_viewer* (1) and *dmilog_viewer* (1). SCR commands require root privilege.

`-o` *`ofile`* The `-o` option causes the filtered output to be written to the output file *`ofile`*. The `-o` option implies the `-n` option described below. If *`ofile`* is `-`, the output is written to stdout. If `-o` is omitted, the output is written to either stdout (if `-n` is specified) or to the interactive `scrlog_viewer` display (if `-n` is omitted).

`-t` The `-t` option enables automatic timestamping. If specified, each log entry is tagged with the time of day at which it was written. Timestamping is disabled by default.

`-n` The `-n` option forces non-interactive behavior. If specified, `scrlog_viewer` runs non-interactively, using the default or specified values for all supported options and source/destination files.

`file` Specifies the name of the file from which log data is read. The format of the data in the specified file must be the same as that used for raw SCR logfile data. If omitted, the SCR logfile is read. If *`file`* is `-`, stdin is read and `scrlog_viewer` runs non-interactively. If given, *`file`* must be the last argument specified on the command line.

EXAMPLES

Capture the current contents of the SCR logfile using default filtering, and put into the file `scr.out`:

```
scrlog_viewer -n > scr.out
```

The following example does the same thing:

```
scrlog_viewer -o scr.out
```

View only the commands executed between 8am June 5, 1999 and 10pm August 14, 1999:

```
scrlog_viewer -s 0605080099 -e 0814220099 -lC
```

Non-interactively read data from stdin, timestamp it, and save the result in the file `stdin.out`:

```
cat datafile | scrlog_viewer -t -o stdin.out
```

FILES

`/var/opt/scr/log/scrlog.log` SCR logfile.

`/var/opt/scr/log/scrlog.old` Archived version of `scrlog.log`, created when the logfile is automatically trimmed by SCR when its size becomes too large. Its contents are included in the log entries read by `scrlog_viewer`.

`/tmp/LFV_<pid>` Temporary files used by `scrlog_viewer`.

`/tmp/LFV_RUN<pid>`

SEE ALSO

scrconfig (1M), *scrdelete* (1M), *scrdiff* (1M), *scrfilter* (1M), *scrhist* (1M), *scrstatus* (1M), *scrtag* (1M), *scrupdate* (1M), *scrviewer* (1M), *scr* (5).

A SCR File Set

This Appendix tells you how to get information on SCR file sets.

SCR is comprised of three filesets: SCR-RUN, SCR-MAN, and SCR-MX. If more information regarding the fileset contents and their attributes is needed, then the following command can be executed:

```
/usr/sbin/swlist -v -l file SCR.SCR-RUN SCR.SCR-MAN \
SCR.SCR-MX
```

This command will display the directories and files for each fileset, and their attributes such as size, owner, group, and type.