Parallel System Support Programs for AIX

Command and Technical Reference, Volume 1

Version 3 Release 2

SA22-7351-02

Parallel System Support Programs for AIX

Command and Technical Reference, Volume 1

Version 3 Release 2

SA22-7351-02

Note!

Before using this information and the product it supports, read the information in “Notices” on page 487.

Third Edition (April 2000)

This edition applies to version 3 release 2 of the IBM Parallel System Support Programs for AIX (PSSP) licensed program (product number 5765-D51) and to all subsequent releases and modifications until otherwise indicated in new editions. This edition replaces SA22-7351-01. Significant changes or additions to the text are indicated by a vertical line (|) to the left of the change.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this publication, or you may address your comments to the following address:

International Business Machines Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States & Canada): 1+914+432-9405
FAX (Other Countries):
Your International Access Code +1+914+432-9405

IBMLink (United States customers only): IBMUSM10(MHVRCFS)
IBM Mail Exchange: USIB6TC9 at IBMMAIL
Internet e-mail: mhvrcfs@us.ibm.com

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this book
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

About This Book

This book provides detailed syntax and parameter information for all commands you can use to install, customize, and maintain the IBM RS/6000 SP system.

For a list of related books and information about accessing online information, see the bibliography in the back of the book.

This book applies to PSSP version 3 release 2. To find out what version of PSSP is running on your control workstation (node 0), enter the following:

```
splst_versions -t -n0
```

In response, the system displays something similar to:

```
0 PSSP-3.2
```

If the response indicates **PSSP-3.2**, this book applies to the version of PSSP that is running on your system.

To find out what version of PSSP is running on the nodes of your system, enter the following from your control workstation:

```
splst_versions -t -G
```

In response, the system displays something similar to:

```
1 PSSP-3.2

2 PSSP-3.2

7 PSSP-3.1.1

8 PSSP-2.4
```

If the response indicates **PSSP-3.2**, this book applies to the version of PSSP that is running on those nodes.

If you are running mixed levels of PSSP, be sure to maintain and refer to the appropriate documentation for whatever versions of PSSP you are running.

Who Should Use This Book

This book is intended for anyone not familiar with the syntax and use of the RS/6000 SP commands.

How This Book Is Organized

This book consists of two volumes. Volume 1 contains RS/6000 SP commands A - R. Volume 2 contains RS/6000 SP commands S - W, RS/6000 SP Files and Other Technical Information, and RS/6000 SP Subroutines. Both volumes share a common frontmatter, appendix, glossary, and bibliography. The indexes are customized for each volume.

Command Format

The commands in this book are in the following format:

Purpose
Provides the name of the command and a brief description of its purpose.

Syntax
Includes a diagram that summarizes the use of the command.

Flags
Lists and describes the options that control the behavior of the command.

Operands
Lists and describes the objects on which the command operates.

Description
Includes a complete description of the command.

Environment Variables
Lists any environment variables that affect the operation of the command. Lists any environment variables that are affected by the operation of the command.

Files
Lists any RS/6000 SP system files that are read, employed, referred to, or written to by the command, or that are otherwise relevant to its use.

Standard Input
Describes what this command reads from standard input.

Standard Output
Describes what this command writes to standard output.

Standard Error
Describes what and when this command writes to standard error.

Exit Values
Describes the values returned and the conditions that caused the values to be returned.

Security
Describes who can run this command and provides other security-related information.

Restrictions
Lists restrictions beyond the security restrictions described previously.

Implementation Specifics
Identifies the package of each individual command.

Prerequisite Information
Provides a pointer to other documents that would enhance the user's understanding of this command.

Location
Specifies the location of the command.

Related Information
Lists RS/6000 SP commands, functions, file formats, and special files that are employed by the command, that have a purpose which is related to that of the command, or that are otherwise of interest within the context of the command. Also listed are related RS/6000 SP documents, other related documents, and miscellaneous information related to the command.

Examples
Provides examples of how the command is typically used.

Typographic Conventions

This book uses the following typographic conventions:

<table>
<tr><th>Typographic</th><th>Usage</th></tr>
<tr><td>Bold</td><td>• Bold words or characters represent system elements that you must use literally, such as commands, flags, and path names.
• Bold words also indicate the first use of a term included in the glossary.</td></tr>
<tr><td>Italic</td><td>• Italic words or characters represent variable values that you must supply.
• Italics are also used for book titles and for general emphasis in text.</td></tr>
<tr><td><code>Constant width</code></td><td>Examples and information that the system displays appear in <code>constant width</code> typeface.</td></tr>
<tr><td>[ ]</td><td>Brackets enclose optional items in format and syntax descriptions.</td></tr>
<tr><td>{ }</td><td>Braces enclose a list from which you must choose an item in format and syntax descriptions.</td></tr>
<tr><td>|</td><td>A vertical bar separates items in a list of choices. (In other words, it means “or.”)</td></tr>
<tr><td>< ></td><td>Angle brackets (less-than and greater-than) enclose the name of a key on the keyboard. For example, <Enter> refers to the key on your terminal or workstation that is labeled with the word Enter.</td></tr>
<tr><td>...</td><td>An ellipsis indicates that you can repeat the preceding item one or more times.</td></tr>
<tr><td><Ctrl-<i>x</i>></td><td>The notation <Ctrl-<i>x</i>> indicates a control character sequence. For example, <Ctrl-c> means that you hold down the control key while pressing <c>.</td></tr>
<tr><td>\</td><td>The continuation character is used in coding examples in this book for formatting purposes.</td></tr>
</table>

Part 1. Command Reference Volume 1

Chapter 1. Commands

This volume contains the RS/6000 SP Commands A - P. See Volume 2 for RS/6000 SP Commands R - W, RS/6000 SP Files and Other Technical Information, and RS/6000 SP Subroutines.

To access the RS/6000 SP online manual pages, set the MANPATH environment variable as follows:

for ksh

```
export MANPATH=$MANPATH:/usr/lpp/ssp/man
```

for csh

```
setenv MANPATH $MANPATH\:/usr/lpp/ssp/man
```

System Partitioning and Commands

When you partition your system, you create one or more system partitions which, for most tasks, function as separate and distinct logical RS/6000 SP systems. Most commands function within the boundary of the system partition in which they are executed. A number of commands, however, continue to treat the RS/6000 SP as a single entity and do not respect system partition boundaries. That is, in their normal function they may affect a node or other entity outside of the current system partition. In addition, some commands which normally function only within the current system partition have been given a new parameter which, when used, allows the scope of that command to exceed the boundaries of the current system partition.

On the control workstation, the administrator is in an environment for *one system partition at a time*. The SP_NAME environment variable identifies the system partition to subsystems. (If this environment variable is not set, the system partition is defined by the `primary:` stanza in the **/etc/SDR_dest_info** file.) Most tasks performed on the control workstation that get information from the System Data Repository (SDR) will get the information for *that particular system partition*.

In managing multiple system partitions, it is helpful to open a window for each system partition. You can set and export the SP_NAME environment variable in each window and set up the window title bar or shell prompt with the system partition name. The following script is an example:

```
sysparenv:

# !/bin/ksh

  for i in 'splst_syspars'

  do

     syspar='host $i | cut -f 1 -d"."'

     echo "Opening the $syspar partition environment"

     sleep 2

     export SP_NAME=$syspar

     aixterm -T "Work Environment for CWS 'hostname -s' - View: $syspar" -ls -sb &

  done

  exit


.profile addition:

# Added for syspar environment setup

  if [ "'env | grep SP_NAME | cut -d= -f1'" = SP_NAME ]

     then

        PS1="[`hostname -s`<p>$SP_NAME] ["'$PWD]> '

     else

        PS1="[`hostname -s`]["'$PWD]> '

  fi

  export ENV
```

As a user, you can check what system partition you're in with the command:

```
spget_syspar -n
```

The following table summarizes those commands which can exceed the boundary of the current system partition. Unless otherwise stated, commands not listed in this table have as their scope the current system partition.

Command	Effect
arp	Can reference any node (by its host name) in any system partition.
Automounter commands	Host names need not be in the current system partition.
chauthpar -p	The **-p** flag allows specification of a system partition other than the current system partition.

Command	Effect
chauthpts -p	The **-p** flag specifies the partition for which the active authentication methods are set.
config_spsec -p	The **-p** flag configures SP Services into the DCE database for servers in the specified partition only.
create keyfiles -p	The **-p** flag creates keyfiles for principals in the specified partition only.
crunacct	Merges accounting data from all nodes regardless of system partition boundaries.
cshutdown -G	The **-G** flag allows specification of target nodes outside of the current system partition.
cstartup -G	The **-G** flag allows specification of target nodes outside of the current system partition.
dsh **dsh -w**{*hostname* | -}	Hosts added to the working collective by host name need not be in the current system partition.
dsh -aG	The **-G** flag modifies the **-a** flag (all nodes in the current system partition) by expanding the scope to all nodes in the entire physical SP system.
Eclock	There is a single switch clock for the SP regardless of the number of system partitions.
Efence -G	The **-G** flag allows specification of nodes outside of the current system partition.
emonctrl -c	The system partition-sensitive control script for the **emon** subsystem supports the **-c** option, which crosses system partitions.
Eunfence -G	The **-G** flag allows specification of nodes outside of the current system partition.
haemctrl -c **haemctrl -u**	The system partition-sensitive control script for the **haem** subsystem supports the **-c** and **-u** options, which cross system partitions.
haemd_SP *syspar_IPaddr*	Specifies the IP address of the system partition in which the **haemd** daemon is to execute.
haemqvar	If the **SP_NAME** environment variable is not set, the default system partition is used.
hagsctrl -c **hagsctrl -u**	The system partition-sensitive control script for the **hags** subsystem supports the **-c** and **-u** options, which cross system partitions.
hatsctrl -c **hatsctrl -u**	The system partition-sensitive control script for the **hats** subsystem supports the **-c** and **-u** options, which cross system partitions.
hmcmds -G	The **-G** flag allows the **hmcmds** commands to be sent to any hardware on the SP system.
hmmon -G	The **-G** flag allows for the specification of hardware outside of the current system partition.
hostlist **hostlist -f** *filename* **hostlist -w** *hostname*	Host names need not be in the current system partition.
hostlist -aG | -nG | -sG	The **-G** flag modifies the **-a**, **-n**, or **-s** flag by expanding the scope to the entire physical SP system.

Command	Effect
hrctrl -c	The system partition-sensitive control script for the **hr** subsystem supports the **-c** option, which crosses system partitions.
hsdatalst -G	The **-G** flag causes the display of HSD information to be for all system partitions.
lppdiff -aG	The **-G** flag modifies the **-a** flag (all nodes in the current system partition) by expanding the scope to all nodes in the entire physical SP system.
lsauthpar -p	The **-p** flag allows specification of a system partition other than the current system partition.
lsauthpts -p	The **-p** flag specifies the partition for which the active authentication methods are to be listed.
nodecond -G	The **-G** flag allows specification of a node outside of the current system partition.
psyslrpt -w *hostnames*	The host names supplied with the **-w** flag can be in any system partition (the **-a** flag will select all nodes in the current system partition).
psyslclr -w *hostnames*	The host names supplied with the **-w** flag can be in any system partition (the **-a** flag will select all nodes in the current system partition).
penotify -w *hostnames*	The host names supplied with the **-w** flag can be in any system partition (the **-a** flag will select all nodes in the current system partition).
pmanctrl -c	The system partition-sensitive control script for the **pman** subsystem supports the **-c** option, which crosses system partitions.
Parallel commands: • **p_cat** • **pcp** • **pdf** • **pfck** • **pexec** • **pexscr** • **pfind** • **pfps** • **pls** • **pmv** • **ppred** • **pps** • **prm**	Parallel commands can take the following options and will behave accordingly: **-w** Host names specified with **-w** need not be in the current system partition. *noderange* Nodes specified by *noderange* must be in the current system partition. *hostlist_args* Host names specified with **hostlist** options **-w** or **-G** need not be in the current system partition (any other **hostlist** options operate within the current system partition).
rm_spsec	The **-p** flag removes configuration from DCE service principals and keyfiles for the specified partition.
SDRArchive, **SDRRestore**	Archives/restores the SDR representing the entire SP.
SDRGetObjects -G	The **-G** flag allows for retrieval of partitioned class objects from partitions other than the current system partition. Without the **-G**, objects which are in a partitioned class are retrieved from the current system partition only.
SDRMoveObjects	Moves objects from one system partition to another.
SDRScan	Scans SDR database files.
SDRValidateString	Checks a character string for valid SDR input.
Other SDR commands	SDR commands that create, change or delete values work within the system partition. Note though that System classes (Frame, for example) are shared among all system partitions. Changes to system classes will affect other system partitions.

Command	Effect
Security commands: • **ext_srvtab** • **kadmin** • **kdb_destroy** • **kdb_edit** • **kdb_init** • **kdb_util** • **k4destroy** • **k4init** • **k4list** • **kpasswd** • **kprop** • **ksrvtgt** • **ksrvutil** • **kstash** • **rcmdtgt** • **setup_authent** • **spseccfg**	The function of these security commands is unchanged under system partitioning. That is, if they previously affected the entire SP, they continue to do so even if the system has been partitioned. If they previously had the ability to affect a remote node that function is unchanged in a system partitioned environment.
sp_configdctrl -c	The system partition-sensitive control script for the **sp_configd** subsystem supports the **-c** option, which crosses system partitions.
spapply_config	Applies a system partition configuration to the entire SP.
spacl -G	Manages ACLs for object instances outside the current partition.
spbootins	If a boot server outside of the current system partition is specified, that node is prepared appropriately.
spbootlist	The command targets nodes in any system partition.
spchvgobj	The command targets nodes in any system partition.
spframe	Configures data for one or more frames across the entire SP.
sphardware	Global system partition can be selected from within the Perspective.
splm	The target nodes defined in the input table can include nodes from any system partition.
splst_versions -G	The **-G** flag allows for retrieval of PSSP version information from nodes outside the current system partition.
splstdata -G	The **-G** flag allows display of information on nodes and adapters outside of the current system partition.
splstadapters -G	The **-G** flag lists information about target adapters outside of the current system partition.
splstnodes -G	The **-G** flag lists information about target nodes outside of the current system partition.
spmirrorvg	The command targets nodes in any system partition.
spmkvgobj	The command targets nodes in any system partition.
spmon -G	The **-G** flag allows specification of nodes outside of the current system partition. The **-G** flag is required when performing operations on any frame or switch.
sprestore_config	Restores the entire SP SDR from a previously made archive.
sprmvgobj	The command targets nodes in any system partition.

Command	Effect
spsitenv	Site environment variables are specified for the SP system as a whole. The specification of **acct_master=** can be any node in the SP regardless of system partition. The specification of **install_image=** may cause boot server nodes outside of the current system partition to refresh the default installation image they will serve to their nodes.
spsyspar	Command is always in global mode.
sptg	Can launch TaskGuides that affect nodes in any system partition.
spunmirrorvg	The command targets nodes in any system partition.
spverify_config	Verifies the configuration of all system partitions in the SPsystem.
st_clean_table	Can specify a node name which is outside the current partition.
st_status	Can specify a node which is outside the current partition.
supper	File collections are implemented and managed without respect to system partition boundaries.
sysctl	The Sysctl client can send requests to any node in the SP.
syspar_ctrl -c -G	The **-c** and **-G** flags allow for the crossing of system partitions in providing a single interface to the control scripts for the system partition-sensitive subsystems.
s1term -G	The **-G** flag allows specification of a node outside of the current system partition.
vsdatalst -G	The **-G** flag causes the display of IBM Virtual Shared Disk information to be for all system partitions.
vsdsklst -G	The **-G** flag specifies the display of information for disks outside the current system partition.

add_principal

Purpose

add_principal – Creates principals in the Kerberos Version 4 authentication database.

Syntax

add_principal [–**r** *realm_name*] [–**v**] *file_name*

Flags

–**r** Adds Kerberos Version 4 principals to a realm other than the local realm.

–**v** Specifies verbose mode. A message is written to standard output for each principal added to the authentication database.

Operands

file_name Specifies the file containing principal names and passwords to add to the Kerberos Version 4 authentication database.

Description

This command provides an interface to the Kerberos Version 4 authentication database to add an entry for a user or service instance, supplying the password used to generate the encrypted private key. The **add_principal** command is suitable for mass addition of users or multiple instances of servers (for example, SP nodes).

This command operates noninteractively if you have a valid ticket-granting-ticket (TGT) for your **admin** instance in the applicable realm. A TGT can be obtained using the **k4init** command. If you do not have a TGT for the **admin** instance for the realm in which you are adding principals, or if the **add_principal** command cannot obtain a service ticket for changing passwords using the **admin** TGT, the user is prompted for the password for the user's **admin** instance.

Administrators use the **add_principal** command to register new users and services instances to the authentication database. An administrator must have a principal ID with an instance of **admin**. Also, *user_name***.admin** must appear in the **admin_acl.add** Access Control List (ACL).

The **add_principal** program communicates over the network with the **kadmind** program, which runs on the machine housing the primary authentication database. The **kadmind** program creates new entries in the database using data provided by this command.

When using the **add_principal** command, the principal's expiration date and maximum ticket lifetime are set to the default values. To override the defaults, the root user must use the **kdb_edit** command to modify those attributes.

Input to the program is read from the file specified by the *file_name* argument. It contains one line of information for each principal to be added, in the following format:

```
name[.instance][@realm] password
```

Note: The @realm cannot be different from the local realm or the realm argument if the **-r** option is specified.

For user entries with a NULL instance, this format matches that of the log file created by the **spmkuser** command. Any form of white space can surround the two fields. Blank lines are ignored. Any line containing a # as the first nonwhite space character, is treated as a comment.

Since the input file contains principal identifiers and their passwords, ensure that access to the file is controlled. You should remove the input file containing the unencrypted passwords after using it, or delete the passwords from it.

The **add_principal** command does not add principals to an AFS authentication database. If authentication services are provided through AFS, use the AFS **kas** command to add principals to the database. Refer to the chapter on security in *PSSP: Administration Guide* for an overview.

Files

/var/kerberos/database/admin_acl.add
Access Control List file.

Exit Values

0 Indicates success. It does not mean that all IDs were added. Individual messages indicate what was added.

nonzero Indicates an error with an appropriate message.

Security

You must be logged in (using **k4init**) as a Kerberos Version 4 database administrator who is authorized to add users. Your Kerberos Version 4 principal must be *your-AIX-username*.**admin** and be listed in the file **/var/kerberos/database/admin_acl.add**.

Location

/usr/lpp/ssp/kerberos/bin/add_principal

Related Information

Commands: **kadmin**, **k4init**, **kpasswd**, **ksrvutil**

Refer to the "RS/6000 SP Files and Other Technical Information" section of *PSSP: Command and Technical Reference* for additional **Kerberos** information.

allnimres

Purpose

allnimres – Allocates Network Installation Management (NIM) resources from a NIM master to a NIM client.

Syntax

allnimres −**h** | −**l** *node_list*

Flags

−**h** Displays usage information. If the command is issued with the −**h** flag, the syntax description is displayed to standard output and no other action is taken (even if other valid flags are entered along with the −**h** flag).

−**l** *node_list* Indicates by *node_list* the SP nodes to which to allocate installation resources. The *node_list* is a comma-separated list of node numbers.

Operands

None.

Description

Use this command to allocate all necessary NIM resources to a client based on the client's **bootp_response** in the System Data Repository (SDR). This includes executing the **bos_inst** command for allocation of the boot resource and nimscript resource. At the end of this command, nodes are ready to netboot to run installation, diagnostics, or maintenance. If the node's **bootp_response** is "disk", all NIM resources are deallocated from the node.

Standard Error

This command writes error messages (as necessary) to standard error.

Exit Values

0 Indicates the successful completion of the command.

−**1** Indicates that an error occurred.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Location

/usr/lpp/ssp/bin/allnimres

Related Information

Commands: **setup_server**, **unallnimres**

Examples

To allocate boot/installation resources to boot/install client nodes 1, 3, and 5 from their respective boot/install servers, enter:

```
allnimres -l 1,3,5
```

arp

Purpose

/usr/lpp/ssp/css/arp – Displays and modifies address resolution.

Syntax

arp {*host_name* | –**a** [**/dev/kmem**]} | –**d** *host_name* |
–**s** *type host_name adapter_address* [*route*] [*temp*] [*pub*] |
–**f** *file_name* [*type*]

Parameters

–**a**
Displays all of the current Address Resolution Protocol (ARP) entries. Use the **crash** command to look at KMEM or UMUnix variables. Specify the –**a /dev/kmem** flag to display ARP information for kernel memory.

–**d** *host_name*
Deletes an ARP entry for the host specified by the *host_name* variable if the user has root user authority.

–**f** *file_name*
Causes the file specified by the *file_name* variable to be read and multiple entries to be set in the ARP tables. Entries in the file should be in the form:

```
type host_name adapter_address [route] [temp] [pub]
```

–**s** *type host_name adapter_address* **[** *route***]** **[***temp***]** **[***pub***]**
Creates an ARP entry for the host specified by the *host_name* variable with the adapter address specified by the *adapter_address* variable. The adapter address is given as 6 hexadecimal bytes separated by colons. The line must be in the following format:

```
type host_name adapter_address [route] [temp] [pub]
```

where:

type
Specifies the type of hardware address as follows:

ether	An Ethernet interface
802.3	An 802.3 interface
switch	A Scalable POWERparallel Switch (SP Switch)
fddi	A Fiber Distributed Data Interface
802.5	A token-ring interface

host_name
Specifies the host_name for which to create an entry.

adapter_address
Specifies the physical address (switch node number) for the switch adapters.

route
Specifies the route for a token-ring interface or Fiber Distributed Data Interface (FDDI) as defined in the token-ring or FDDI header.

temp
Specifies that this ARP table entry is temporary. The table entry is permanent if this argument is omitted.

pub Specifies that this table entry is to be published, and that this system acts as an ARP server responding to requests for *host_name*, even though the host address is not its own.

Description

The **arp** command has been modified to add support for the switch. This command is valid only on an SP system.

The **arp** command displays and modifies the Internet-to-adapter address translation tables used by ARP. The **arp** command displays the current ARP entry for the host specified by the *host_name* variable. The host can be specified by name or number, using Internet dotted decimal notation.

Related Information

SP Command: **ifconfig**

AIX Commands: **crash**, **netstat**

AIX Daemon: **inetd**

Refer to *PSSP: Administration Guide* for additional information on the SP Switch.

Refer to "TCP/IP Protocols" in *AIX Version 4.1 System Management Guide: Communications and Networks*.

Examples

1. To add a single entry to the **arp** mapping tables until the next time the system is restarted, enter:

```
arp -s switch host2 1
```

2. To delete a map table entry for the specified host with the **arp** command, enter:

```
arp -d host1
```

cfghsd

Purpose

cfghsd – Configures a hashed shared disk.

Syntax

cfghsd {–**a** | *hsd_name* ...}

Flags

–**a** Specifies all the hashed shared disks that have been defined.

Operands

hsd_name Specifies a defined hashed shared disk. All underlying virtual shared disks in the hashed shared disk must be configured before using this command.

Description

This command configures the already defined hashed shared disks and makes them available. The command extracts information from the System Data Repository (SDR).

Security

You must be in the AIX **bin** group to run this command.

Restrictions

If you have the IBM Recoverable Virtual Shared Disk software installed and operational, do not use this command. The results may be unpredictable.

See *PSSP: Managing Shared Disks*.

Prerequisite Information

PSSP: Managing Shared Disks

Location

/usr/lpp/csd/bin/cfghsd

Related Information

Commands: **defhsd**, **hsdatalst**, **lshsd**, **ucfghsd**

Examples

To make the hashed shared disk *hsd1* available, enter:

```
cfghsd hsd1
```

cfghsdvsd

Purpose

cfghsdvsd – Configures a hashed shared disk and the underlying virtual shared disks that comprise it and starts the virtual shared disks.

Syntax

cfghsdvsd −**a** | {*hsd_name*...}

Flags

−**a** Specifies that all the hashed shared disks defined on this system or system partition are to be configured (made available).

Operands

hsd_name Specifies the names of defined hashed shared disks that are to be configured. This command configures the underlying virtual shared disks as well.

Description

Use this command to configure already-defined hashed shared disks and their underlying virtual shared disks and make them available. Note all of the virtual shared disks go to the active state, making them available to the node on which this command is run.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit hsd_mgmt
```

and select the Configure a hashed shared disk and its underlying virtual shared disks option.

Security

You must have access to the virtual shared disk subsystem via the sysctl service to run this command.

Prerequisite Information

PSSP: Managing Shared Disks

Location

/usr/lpp/csd/bin/cfghsdvsd

Related Information

Commands: **cfghsd**, **cfgvsd**, **ucfghsdvsd**

Examples

To configure the hashed shared disk *hsd1* and the virtual shared disks that comprise it, enter:

```
cfghsdvsd hsd1
```

cfgvsd

Purpose

cfgvsd – Configures a virtual shared disk.

Syntax

cfgvsd {**–a** | *vsd_name* ...}

Flags

–a Specifies all virtual shared disks that have been defined.

Operands

vsd_name Specifies a defined virtual shared disk.

Description

Use this command to configure the already defined virtual shared disks and bring them to the stopped state. It does not make the virtual shared disk available. The command extracts information from the System Data Repository (SDR).

You can use the System Management Interface Tool (SMIT) to run the **cfgvsd** command. To use SMIT, enter:

```
smit vsd_mgmt
```

and select the Configure a virtual shared disk option.

Security

You must be in the AIX **bin** group to run this command.

Restrictions

If you have the IBM Recoverable Virtual Shared Disk software installed and operational, do not use this command. The results may be unpredictable.

See *PSSP: Managing Shared Disks*.

Prerequisite Information

PSSP: Managing Shared Disks

Location

/usr/lpp/csd/bin/cfgvsd

Related Information

Commands: **ctlvsd**, **lsvsd**, **preparevsd**, **resumevsd**, **startvsd**, **stopvsd**, **suspendvsd**, **ucfgvsd**

Examples

To bring the virtual shared disk **vsd1vg1n1** from the defined state to the stopped state, enter:

```
cfgvsd vsd1vg1n1
```

chauthpar

Purpose

chauthpar – Enables the active remote command authentication methods for a system partition.

Syntax

chauthpar {−**h** | [−**c** | −**f**] [−**p** *partition*] [−**v**] *method*...}

Flags

−h Specifies that the command is only to display the valid command syntax. When this flag is specified, other flags and operands are ignored.

−c Specifies that the command is to operate only on the control workstation, changing settings in the System Data Repository (SDR) and in AIX as required, without attempting to make any changes on the nodes in the partition. Use this flag when you are activating a method that is newly configured on the control workstation but not yet configured on the nodes.

−f Specifies that the command is to attempt to change the setting on all accessible nodes in the partition, even when no change is indicated to the SDR or to the AIX setting on the control workstation.

−p *partition* Specifies the partition for which the active authentication methods are to be set. The partition can be specified in either hostname or IP address format. If none is specified, the value of the **SP_NAME** environment variable will be used. If **SP_NAME** is not set in this case, the default partition is assumed.

−v Specifies verbose output − informational messages are to be displayed.

Operands

method Specifies an available authentication method. When you specify multiple authentication methods, they must be specified in the order shown, which is the order of precedence in their use by the remote commands.

You must specify at least one of the following authentication methods that are used by the remote commands.

k5 Specifies that the Kerberos Version 5 authentication method is to be made active for this partition. To activate **k5**, you must have previously issued **spsetauth** −**i** to select DCE capability for the partition.

k4 Specifies that the Kerberos Version 4 authentication method is to be made active for this partition. To activate **k4**, you must have previously issued **spsetauth** −**i** to select **k4** capability for the partition.

std Specifies that the Standard AIX authentication method is to be made active for this partition. If specified, this method must be last (lowest priority).

Description

The **chauthpar** command enables the specified authentication methods for the designated system partition. All methods not included are set inactive (not in use).

You can limit operation of the command to the control workstation by specifying the −**c** option. Conversely, the −**f** option allows you to force the setting to be propagated to all accessible nodes in the partition, regardless of whether it was changed. Normal command execution, with neither option, propagates any changed setting to the nodes.

This command should be the sole vehicle for managing the authentication methods settings on the SP. Should the settings become damaged due to system problems or inappropriate use of SDR interfaces, this command will detect erroneous settings and inform the user of them. When a system partition is found with an incorrect setting for the remote command authentication methods, it will be reset using the following rules:

- Sets **std**, if Standard AIX is active locally on the control workstation.
- Sets **k4**, if Kerberos 4 AIX is active locally on the control workstation and **k4** was set for the partition using **spsetauth** −**i**.
- Sets **k5**, if Kerberos 5 is active locally on the control workstation and **dce** was set for the partition using **spsetauth** −**i**.

These automatic corrections to other partitions are made only in the SDR and are not propagated to any nodes. If this occurs, you should examine the changed settings and reissue this command against each of the affected partitions to complete the repair. If the change made to the SDR when the error was detected is correct, reissue the command with the −**f** option to insure that all nodes have the new setting. If the reset value is not correct for the partition, reissue the command without −**f** to change it as required.

Because the propagation of settings to running nodes is performed using the AIX **rsh** command to execute **chauthent**, the control workstation and the nodes must have at least one common remote command authentication method active in order for propagation to succeed. When this in not the case, propagation can only be completed by the local root user running **chauthent** or **spauthconfig** on each node (or by a re-boot). The same applies to nodes that are not running or are otherwise inaccessible when this command is executed.To activate the Kerberos 5 authentication method in a partition, you must have configured the control workstation and the partition for DCE using the commands or SMIT panels. To activate the Kerberos Version 4 authentication method, you must have configured the control workstation and the partition for k4, using the commands or SMIT panels. You must activate the Kerberos Version 4 authentication method if any node in the partition is running a level of PSSP earlier than Version 3.2.

Consequences of Error

A problem executing the **chauthent** command remotely on some or all nodes in the partition does not result in unsuccessful execution of the command. Error messages from the **dsh** and **rsh** commands should be noted to determine the reason for each problem. For example, when the **−c** flag is not specified but no nodes in the partition are running, the **dsh** command will be unsuccessful and return the message:

```
dsh: 5025–511 No hosts in working collective
```

Establishing the correct settings on inaccessible nodes will require the root user to run **chauthent** or **spauthconfig** on those nodes (or re-boot them).

Environment Variables

The **SP_NAME** variable can be used to designate the applicable partition.

Standard Output

Output consists of informational messages, when the **−v** option is specified.

Standard Error

Output consists of error messages when the command cannot complete successfully. Even when the command returns **0**, error messages will be output if defective SDR attributes were repaired for another partition or propagation to one or more nodes was unsuccessful.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error occurred.

Security

You must have root privilege to run this command.

Restrictions

The **chauthpar** command may be executed only on the control workstation.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP) (fileset **ssp.clients**).

Prerequisite Information

The chapters on security in the *PSSP: Administration Guide*.

Location

/usr/lpp/ssp/bin/chauthpar

Related Information

Commands: **lsauthpar**

Examples

1. To set all remote command methods active in a partition, enter:

```
$chauthpar -p sp3b k5 k4 std
```

2. To set the current partition (SP_NAME assumed to be set) to use the Kerberos 4 method, and the Standard AIX method, enter:

```
$chauthpar k4 std
```

3. Setting Kerberos 4 and Kerberos 5 and Standard AIX as the active authentication methods for remote commands running in a partition, with detailed output:

```
$chauthpar -v -p 9.10.11.12 k5 k4 std

The remote command authentication methods for this host are currently

k4:std

The authentication methods by partition are currently

abcsp1  k4:std

abcsp2  k4:std

abcsp3  k4:std

The partition to be modified is abcsp3

The auth_methods attribute of the partition has been set to k5:k4:std

The remote command authentication methods for this host are now set to

k5:k4:std

The chauthent command was executed successfully on node s1n15.abc.org

The chauthent command was executed successfully on node s1n16.abc.org

The chauthent command was executed successfully on node s1n18.abc.org

The chauthent command was executed successfully on node s1n19.abc.org
```

chauthpts

Purpose

chauthpts – Enables the active trusted services authentication methods for a system partition.

Syntax

chauthpts {−**h** | [−**c** | −**f**] [−**p** *partition*] [−**v**] [*method* ...]}

Flags

−**h** Specifies that the command only display the valid command syntax. When −**h** is specified, all other flags and operands are ignored.

−**c** Specifies that the command operate only on the control workstation, changing System Data Repository (SDR) and AIX settings as required, without making any changes on the nodes in the partition. Use −**c** when you are activating a method that is newly configured on the control workstation, but not yet configured on the nodes.

−**f** Specifies that the command change the settings on all accessible nodes in the partition, even when no change is indicated to the SDR or AIX settings on the control workstation.

−**p** *partition*
Specifies the partition for which the active authentication methods are set. The partition can be specified in either hostname or IP address format. If none is specified, the value of the **SP_NAME** environment variable is used. If **SP_NAME** is not set in this case, the default partition is assumed.

−**v** Specifies verbose output. Informational messages are displayed

Operands

method Specifies an available authentication method. When you specify multiple authentication methods, they must be specified in the order shown, which is the order of precedence in their use by the trusted services. You may choose any combination of the following authentication methods. These authentication methods are used by various trusted services.

Note: Choosing no method makes all authentication methods inactive for the specified partition.

dce Specifies that the DCE authentication method is to be made active for this partition. To activate DCE, you must have previously issued **spsetauth** −**i** to select DCE capability for the partition.

compat Specifies that the security mechanisms used by trusted services in prior releases are to be made active. For more information on the security mechanisms see *PSSP: Administration Guide*. To activate *compat*, you must have previously issued **spsetauth** −**i** to select **k4** capability for the partition. If specified, this method must be last (lowest priority).

Description

The **chauthpts** command enables authentication methods for the designated system partition. All methods not included are set inactive (not in use).

You can limit operation of the command to the control workstation by specifying the −**c** option. The −**f** option forces the setting to be propagated to all accessible nodes in the partition, whether it was changed or not. Using **chauthpts** with neither flag propagates any changed setting to the nodes.

This command should be the sole vehicle for managing the trusted services authentication methods settings on the SP.

Should the settings become corrupted due to system problems or inappropriate use of SDR interfaces, this command will detect erroneous settings and inform the user of them. When a system partition is found with an invalid setting for the trusted services authentication methods, it will be reset using the following rules:

- Set **compat** if **Compatibility** is active locally on the control workstation and **k4** was set for the partition using **spsetauth** −**i**.
- Set **dce** if DCE is active locally on the control workstation and **dce** was set for the partition using **spsetauth** −**i**.

These automatic corrections (to other partitions) are made only in the SDR and are not propagated to any nodes. If this occurs, you should examine the changed settings and reissue this command against each of the affected partitions to complete the repair. If the change made to the SDR when the error was detected is correct, reissue the command with the −**f** option to ensure that all nodes have the new setting. If the reset value is not correct for the partition, reissue the command without −**f** to change it as required.

Because the propagation of settings to running nodes is performed using the AIX **rsh** command to execute **chauthts**, the control workstation and the nodes must have at least one common remote command authentication method active for propagation to succeed. When this is not the case, propagation can only be completed by the local root user running **chauthts** or **spauthconfig** on each node (or by a re-boot). The same applies to nodes that are not running or are otherwise inaccessible when this command is executed.

To activate **DCE** you must configure **DCE**.

To activate the **Compatibility** authentication method, you must have configured the control workstation and the partition for Kerberos Version 5 using the commands or SMIT panels. You must activate the **Compatibility** authentication method if any node in the partition is running a level of PSSP earlier than 3.2.

Environment Variables

The **SP_NAME** variable can be used to designate the applicable partition.

Standard Output

Output consists of informational messages when the −**v** option is specified.

Standard Error

Output consists of error messages, when the command cannot complete successfully. Even when the command returns 0, error messages will be output if a defective SDR attribute was repaired for another partition or propagation to one or more nodes failed.

Unsuccessful remote execution of the **chauthts** command on some or all nodes in the partition does not result in complete command error. Error messages from the **dsh** and **rsh** commands should be noted to determine the reason for each error. For example, when the −**c** flag is not specified but no nodes in the partition are running, the **dsh** command is unsuccessful and returns the message:

```
dsh: 5025−511 No hosts in working collective
```

Establishing the correct settings on inaccessible nodes will require the root user to run **chauthts** or **spauthconfig** on those nodes (or re−boot them).

Exit Values

0 Indicates successful completion of the command.

1 Indicates that an error occurred.

Security

You must have root privilege to run this command.

Restrictions

The **chauthpts** command may be executed only on the control workstation.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP) (fileset **ssp.clients**).

Prerequisite Information

The chapters on security in the *PSSP: Administration Guide*.

Location

/usr/lpp/ssp/bin/chauthpts

Related Information

Commands: **lsauthpts**

Examples

1. To activate all trusted services authentication methods in a partition, enter:

   ```
   $chauthpts -p sp3b dce compat
   ```

2. To set DCE as the sole active authentication method for trusted services running in a partition, without propogating the change to active nodes, enter:

   ```
   $chauthpts -c -p 9.10.11.12 dce
   ```

chauthts

Purpose

chauthts – Enables the active authentication methods for trusted services on a host.

Syntax

chauthts {−**h** | [*method* ...]}

Flags

−h Specifies that the command should only show its syntax. When this flag is specified, any operands are ignored.

Operands

method Specifies an authentication method to be activated. Any methods not specified are made inactive. When you specify multiple authentication methods, they must be specified in the order shown, which is the order of precedence in their use by the trusted services.

You may choose any combination of the following authentication methods. These authentication methods are used by various trusted services.

Note: Choosing no method makes all authentication methods inactive.

dce Specifies that the DCE authentication method is to be made active for this host

compat Specifies that the trusted services that used other security mechanisms in prior releases are to use those methods. If specified, this method must be last (lowest priority).

To activate the **DCE** authentication method, you must have configured the system for DCE. To activate the **Compatibility** authentication method, you must have configured the system for K4.

Description

The **chauthts** command enables the authentication methods used by trusted services on the local host. Trusted services that support multiple methods will attempt to authenticate and authorize client requests using the methods in the order shown. Use this command to set the authentication methods on the control workstation at initial installation and on independent workstations. Use **chauthpts** instead to set the authentication methods for SP nodes (or the control workstation after initial installation).

Standard Output

The local settings are stored in **/spdata/sys1/spsec/auth_methods**.

Standard Error

Output consists of error messages, when the command cannot complete successfully.

If the command is unable to write the new settings into the file, it will attempt to remove the file to avoid the use of spurious information that might compromise security. This will disable all trusted services authentication methods on the local system.

Exit Values

0 Indicates successful completion of the command.

1 Indicates that an error occurred.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP) (fileset **ssp.clients**).

Prerequisite Information

The chapters on security in the *PSSP: Administration Guide*.

Location

/usr/lpp/ssp/bin/chauthts

Related Information

Commands: **lsauthts**

Examples

To set all available authentication methods active, enter:

```
#chauthts dce compat
```

chgcss

Purpose

chgcss – Applies configuration changes to a Scalable POWERparallel Switch (SP Switch or SP Switch2) Communications Adapter.

Syntax

chgcss –**l** *name* {–**a** *dev_memory_alloc* [–**a** *dev_memory_alloc*] | –**a** *win_resource_alloc*}

Flags

–**l** *name*
Specifies the device logical name in the Customized Devices object class whose attribute values should be changed.

–**a** *dev_memory_alloc*
Identifies the device memory attribute to be changed and the value to which it should be changed; *dev_memory_alloc* can be one of the following:

win_poolsize=*size*
Also referred to as "device memory", **win_poolsize** is the total, maximum amount of pinned system memory (in bytes) that can be used as interface network FIFO buffers for SP switch adapter windows.

win_minsize=*size*
The guaranteed minimum amount of device memory (in bytes) per SP switch adapter window; **win_minsize** ensures that all tasks in a job have the minimum required device memory to run.

win_maxsize=*size*
The maximum amount of device memory (in bytes) per SP switch adapter window, **win_maxsize** internally limits device memory usage for each window; **win_maxsize** is further bounded by available device memory and by any job-scheduler specified limit.

rpoolsize=*size*
Size of the IP receive buffer pool (in bytes).

spoolsize=*size*
Size of the IP send buffer pool (in bytes).

Implementation Specifics

Device memory attribute sizes can be specified in decimal or hexadecimal by preceding the value with "0x".

SP Switch configuration changes to the **spoolsize** and **rpoolsize** attributes are later applied to the device when it is configured at system reboot.

–**a** *win_resource_alloc*
Specifies the window resource attribute to reserve, release, or query SP switch adapter windows; *win_resource_alloc* can be one of the following:

window=cmd:reserve/id:*client***/type:***client_type***/count:***requested_count*

window=cmd:release/id:*client*

window=cmd:query/id:*client*

window=cmd:query/id:AVAIL

window=cmd:query/id:

window=cmd:query

where:

client is the application for which windows should be reserved, released, or queried. Use a case-sensitive string to specify the client application. The literal string **AVAIL** is used to indicate an unreserved window.

client_type is reserved for future use. You must specifiy **user_client** for *client_type*.

requested_count is the requested number of windows to reserve for the specified application.

Implementation Specifics

If you have already reserved windows for an application, you cannot reserve additional windows.

The **release** command of the **chgcss window** attribute releases **ALL** windows for the specified client.

Operands

None.

Description

Use this command to change the device memory or window resource allocations for the SP Switch or SP Switch2 Communications Adapter.

The SP Switch adapter multiplexes between independent data streams, where a data stream is represented by an adapter "window." A subset of adapter windows are for system use only (the IP window, for example), while others can be reserved for long-running subsystems such as VSD. Windows that are neither held for system use nor otherwise reserved may be allocated dynamically for large-scale parallel applications.

Files

/var/adm/SPlogs/css/chgcss.log
The log of **chgcss** reserve, release, and query events.

Standard Output

A space-separated list of window numbers or client ID strings is written to standard output upon successful execution of **chgcss** to reserve, release, or query windows. The command also writes informational messages to standard output whenever a device memory attribute is changed.

Standard Error

This command writes error messages to standard error.

Security

You must have root privilege to run this command.

Prerequisite Information

For additional information on values for the device memory attributes, refer to the tuning information at http://www.rs6000.ibm.com/support/sp.

Location

/usr/lpp/ssp/css/chgcss

Related Information

AIX Command: **lsattr**

Examples

1. To change the maximum window size to 1 megabyte, enter:

   ```
   # chgcss -l css0 -a win_maxsize=0x100000

   chgcss: attribute win_poolsize value = 5242880.

   chgcss: attribute win_maxsize value = 1048576.

   chgcss: attribute win_minsize value = 1048576.

   #
   ```

   win_poolsize, **win_minsize** and **win_maxsize** are inter-dependent attributes; all are displayed whenever one or more is changed.

2. To change the size of the IP send and receive buffers to 1 megabyte, enter:

   ```
   # chgcss -l css0 -a rpoolsize=1048576 -a spoolsize=1048576

   chgcss: attribute rpoolsize value = 1048576.

   chgcss: attribute spoolsize value = 1048576.

   #
   ```

   The new values for the **rpoolsize** and **spoolsize** attributes are displayed.

3. To reserve a window for GPFS, enter:

```
# chgcss -l css0 -a window=cmd:reserve/id:GPFS/type:user_client/count:1

1

# chgcss -l css0 -a window=cmd:QUERY/id:GPFS

1

#
```

In this example, window 1 has been reserved for GPFS.

4. To query the list of reserving applications for all windows, enter:

```
# chgcss -l css0 -a window=cmd:query

VSD GPFS AVAIL AVAIL AVAIL

#
```

In this example, window 0 is reserved for VSD, window 1 is reserved for GPFS, and windows 2, 3 and 4 are unreserved.

5. To query the list of unreserved windows, enter:

```
# chgcss -l css0 -a window=cmd:QUERY/id:AVAIL

2 3 4
```

or enter:

```
# chgcss -l css0 -a window=cmd:QUERY/id:

2 3 4
```

In this example, windows 2, 3 and 4 are unreserved.

6. To release windows reserved by GPFS, enter:

```
 # chgcss -l css0 -a window=cmd:RELEASE/id:GPFS

VSD AVAIL AVAIL AVAIL AVAIL

#
```

In this example, window 0 is the only remaining reserved window after the GPFS window is released.

chkp

Purpose

chkp – Changes Kerberos Version 4 principals.

Syntax

chkp −**h**

chkp [−**e** *expiration*] [−**l** *lifetime*] *name*[.*instance*] ...

Flags

−**h**
Displays usage information.

−**e** *expiration*
Specifies a new expiration date for the principals. The date must be entered in the format *yyyy-mm-dd*, and the year must be a value from 1970 to 2037. The time of expiration is set to 11:59 PM local time on the date specified.

−**l** *lifetime*
Specifies the new maximum ticket lifetime for the principals. The lifetime must be specified as a decimal number from 0 to 255. These values correspond to a range of time intervals from five minutes to 30 days. Refer to *PSSP: Administration Guide* for a complete list of the possible ticket lifetime values you can enter and the corresponding durations in days, hours, and minutes. The following list shows a representative sample with approximate durations:

```
lifetime operand - Approximate duration

      141                 1 day

      151                 2 days

      170                 1 week

      180                 2 weeks

      191                 1 month
```

At least one flag must be specified.

Operands

name[.*instance*] **...**
Identifies the principals to change.

Description

Use this command to change principals in the local Kerberos database. It allows the current expiration date and maximum ticket lifetime to be redefined. It cannot be used to change the principal's password. To do that, the administrator must use the **kpasswd**, **kadmin**, or **kdb_edit** commands. The **chkp** command should normally be run only on the primary server. If there are secondary authentication servers, the **push-kprop** command is invoked to propagate the change to the other servers.

The command can be used to update a secondary server's database, but the changes may be negated by a subsequent update from the primary.

Files

/var/kerberos/database/admin_acl.mod

/var/kerberos/database/principal.*
Kerberos database files.

Exit Values

0 Indicates the successful completion of the command. Specified principals that exist were changed. If any principal that you specify does not exist in the database, a message is written to standard error and processing continues with any remaining principals.

1 Indicates that an error occurred and no principal was changed. One of the following conditions was detected:

- The command was incorrectly specified with no operand or no flag, or a flag that is not valid.
- The host on which the command was issued is not an authentication server.
- The **kdb_edit** or **kdb_util** command was unsuccessful.

Security

The **chkp** command can be run by the root user logged in on a Kerberos server host. It can be invoked indirectly as a Sysctl procedure by a Kerberos database administrator who has a valid ticket and is listed in the **admin_acl.mod** file.

Location

/usr/kerberos/etc/chkp

Related Information

Commands: **kadmin**, **kdb_edit**, **lskp**, **mkkp**, **rmkp**, **sysctl**

Examples

1. To set the default maximum ticket lifetime for new principals to (approximately) one week, enter:

   ```
   chkp -l 171 default
   ```

2. To set the maximum ticket lifetime to approximately three weeks and the expiration date to 30 June 2003 for several principals, enter:

   ```
   chkp -l 181 -e 2003-06-30 franklin jtjones root.admin susan
   ```

cksumvsd

Purpose

cksumvsd – Views and manipulates the IBM Virtual Shared Disk component's checksum parameters.

Syntax

cksumvsd [–**s**] [–**R**] [–**i** | –**I**]

Flags

–s Shows IP checksum counters only.

–R Resets IP checksum counters.

–i Calculates IP checksum on all IBM Virtual Shared Disk remote messages.

–I Indicates not to calculate IP checksum on all IBM Virtual Shared Disk remote messages.

If no flags are specified, the current setting of all IBM Virtual Shared Disk checksum parameters and counters are displayed.

Operands

None.

Description

The IBM Virtual Shared Disk IP device driver can calculate and send checksums on remote packets it sends. It also can calculate and verify checksums on remote packets it receives. The **cksumvsd** command is used to tell the device driver whether to perform checksum processing. The default is no checksumming.

Issuing **cksumvsd –i** turns on checksumming on the node on which it is run. **cksumvsd –i** must be issued on all virtual shared disk nodes in the system partition, or the IBM Virtual Shared Disk software will stop working properly on the system partition. If node A has **cksumvsd –i** (checksumming turned on) and node B has **cksumvsd –I** (checksumming turned off, the default), then A will reject all messages from B (both requests and replies), since A's checksum verification will be unsuccessful on all B's messages. The safe way to run **cksumvsd –i** is to make sure that all virtual shared disks on all nodes are in the STOPPED or SUSPENDED states, issue **cksumvsd –i** on all nodes, then resume the needed virtual shared disks on all nodes.

In checksumming mode, the IBM Virtual Shared Disk IP device driver keeps a counter of the number of packets received with good checksums, and the number received with problem checksums. **cksumvsd** and **statvsd** both display these values (**statvsd** calls **cksumvsd –s**).

cksumvsd dynamically responds to the configuration of the IBM Virtual Shared Disk IP device driver loaded in the kernel. Its output and function may change if the IBM Virtual Shared Disk IP device driver configuration changes.

Files

/dev/kmem **cksumvsd** reads and writes **/dev/kmem** to get information to and from the IBM Virtual Shared Disk IP device driver in the kernel.

Security

You must be in the AIX **bin** group to run this command.

You must have write access to the SDR to run this command.

Prerequisite Information

PSSP: Managing Shared Disks

Related Information

Command: **cfgvsd**

Examples

1. To display the IBM Virtual Shared Disk checksum settings and counter values, enter:

   ```
   cksumvsd
   ```

   You should receive output similar to the following:

   ```
   VSD cksum: current values:

   do_ip_checksum: 0

   ipcksum_cntr:   350 good,       0 bad,  0 % bad.
   ```

   The IBM Virtual Shared Disk checksumming is currently turned off on the node. Prior to this, checksumming was turned on and 350 IBM Virtual Shared Disk remote messages were received, all with good checksumming.

2. To turn IBM Virtual Shared Disk checksumming on and display counters, enter:

   ```
   cksumvsd -i
   ```

   You should receive output similar to the following:

   ```
   VSD cksum: current values:

   do_ip_checksum: 0

   ipcksum_cntr:   350 good,       0 bad,  0 % bad.

   VSD cksum: new values:

   do_ip_checksum: 1

   ipcksum_cntr:   350 good,       0 bad,  0 % bad.
   ```

   The command displays old and new values. As before, the node has received 350 IBM Virtual Shared Disk remote messages with good checksums.

3. To display only the IBM Virtual Shared Disk checksum counters, enter:

   ```
   cksumvsd -s
   ```

   You should receive output similar to the following:

```
ipcksum_cntr:   350 good,       0 bad,  0 % bad.
```

cmonacct

Purpose

cmonacct – Performs monthly or periodic SP accounting.

Syntax

cmonacct [*number*]

Flags

None.

Operands

number Specifies which month or other accounting period to process. The default is the current month.

Description

The **cmonacct** command performs monthly or periodic SP system accounting. The intervals are set in the **crontab** file. You can set the **cron** daemon to run the **cmonacct** command once each month or at some other specified time period. By default, if accounting is enabled for at least one node, **cmonacct** executes on the first day of every month.

The **cmonacct** command creates summary files under the **/var/adm/cacct/fiscal** directory and restarts summary files under the **/var/adm/cacct/sum** directory, the cumulative summary to which daily reports are appended.

Security

You must have root privilege to run this command.

Location

/usr/lpp/ssp/bin/cmonacct

Examples

1. To produce reports for the current month, enter:

   ```
   cmonacct
   ```

2. To produce reports for fiscal period 12, enter:

   ```
   cmonacct 12
   ```

config_spsec

Purpose

config_spsec – Configures SP Services into the DCE database. Services which use DCE as an authentication method are required to have certain information entered in the CDS registry and Security Server to perform client/server authentication.

Syntax

config_spsec [–**h**] [–**v**] [–**c** | –**p** *partition_name*]

Flags

–h Prints the command syntax to standard output.

–v Prints progress messages to standard output.

–c Configures only the principals specific to the control workstation. This flag is required when running this command prior to node number being available in the ODM.

–p *partition_name*
Configures service principals specific to partition *partition_name* for the control workstation.

Operands

None.

Description

The **config_spsec** command enters data into the CDS registry and Security Server database. You must be logged into DCE with cell administration authority to use the command. These files contain the information necessary for each service to be configured to use DCE authentication. The command reads from two files which specify groups, service principals and members. The **spsec_defaults** file is shipped with the product and should not be altered by users. The **spsec_overrides** file is provided to allow users to modify principal, groups, and organization names. The program reads the two files, and creates all the necessary entries in the CDS registry and Security Server. If the information is already present, an appropriate message will be issued and logged into the log file (not an error).

For syntax errors within either file, an error message will be issued and logged and processing will halt. Processing of both files occurs prior to any changes being made to any DCE database.

The command prompts for an ID with cell administrator authority, which will be added to the spsec–admin group. The command also prompts for a password. Since the user is required to be logged into the DCE cell as an administrator, the password is that of a cell administrator. This password is required by the config.dce program (called from within this program).

Files

input: **/usr/lpp/ssp/config/spsec_defaults**

/spdata/sys1/spsec/spsec_overrides

output:

/var/adm/SPlogs/auth_install/log

CDS registry and Security Server database updated

Exit Values

0 Indicates successful completion of the command.

1 Indicates that errors occurred during the execution of this program. Review any reported errors either on the console or in the Log file.

An unsuccessful run of this command (depending on where it encountered a problem) may leave the state of service principals in an incomplete state. Some service principals, groups, and directories may not be created or updated. This will cause services to not operate correctly in a DCE environment.

Security

Users need to be logged into the cell with cell administrator authority.

Restrictions

The command runs only on the control workstation and non-SP workstations.

Location

/usr/lpp/ssp/bin/config_spsec

Related Information

File: **/usr/lpp/ssp/config/spsec_defaults**

File: **/spdata/sys1/spsec/spsec_overrides**

Commands: **rm_spsec**

DCE Administration Publications for AIX.

Examples

1. To configure all service principals and accounts, and to set an initial key for each service as a DCE ID with cell administrator authority, enter:

```
config_spsec -v
```

2. To configure control workstation only services (required when the SDR is not available during an initial install) as a DCE ID with cell administrator authority, enter:

```
config_spsec -v -c
```

3. To configure partition *my_par* services, enter:

```
config_spsec -v -p my_par
```

cprdaily

Purpose

cprdaily – Creates an ASCII report of the previous day's accounting data.

Syntax

cprdaily [–**c**] [[–**l**] [*yyyymmdd*]]

Flags

–**c**
Reports exceptional resource usage by command. This flag may be used only on the current day's accounting data.

–**l**
Reports exceptional usage by login ID for the specified date specified in *mmdd* variable, if other than current day's reporting is desired. (This is lowercase **l**, as in **l**ist.)

Operands

yyyymmdd
Specifies the date for exceptional usage report if other than the current date.

Description

This command is called by the **crunacct** command to format an ASCII report of the previous day's accounting data for all nodes. The report resides in the **/var/adm/cacct/sum/rprt***yyyymmdd* file, where *yyyymmdd* specifies the year, month, and day of the report.

Security

You must have root privilege to run this command.

Location

/usr/lpp/ssp/bin/cprdaily

Examples

1. To print usual daily accounting reports (Daily Report, Daily Usage Report, Daily Command Summary, Monthly Total Command Summary, Last Login Report), enter:

   ```
   cprdaily
   ```

2. To print a Command Exception and Login Exception Report, enter:

   ```
   cprdaily -c -l
   ```

3. To print a Login Exception Report for March 16, 1994, enter:

   ```
   cprdaily -l 19940316
   ```

cptuning

Purpose

cptuning – Copies a file to **/tftpboot/tuning.cust**.

Syntax

cptuning −**h** | *file_name*

Flags

−**h**
: Displays usage information for this command (syntax message). If the command is issued with the −**h** flag, the syntax description is displayed to standard output and no other action is taken (even if other valid flags are entered along with the −**h** flag).

Operands

file_name
: Specifies the name of a file to copy to **/tftpboot/tuning.cust**. If the *file_name* begins with a slash (/), the name is considered to be a fully qualified file name. Otherwise, the file name is considered to be in the **/usr/lpp/ssp/install/config** directory.

Description

Use this command to copy the specified file to the **/tftpboot/tuning.cust** file. IBM ships the following four predefined tuning parameter files in **/usr/lpp/ssp/install/config**:

tuning.development
: Contains initial performance tuning parameters for a typical development system.

tuning.scientific
: Contains initial performance tuning parameters for a typical scientific system.

tuning.commercial
: Contains initial performance tuning parameters for a typical commercial system.

tuning.default
: Contains initial performance tuning parameters for a general SP system.

This command is intended for use in copying one of these files to **/tftpboot/tuning.cust** on the control workstation for propagation to the nodes in the SP. It can also be used on an individual node to copy one of these files to **/tftpboot/ tuning.cust**.

Files

Upon successful completion, the **/tftpboot/tuning.cust** file is updated.

Standard Output

When the command completes successfully, a message to that effect is written to standard output.

Standard Error

This command writes error messages (as necessary) to standard error.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error occurred.

If the command does not run successfully, it terminates with an error message and a nonzero return code.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Location

/usr/lpp/ssp/bin/cptuning

Related Information

SP Files: **tuning.commercial**, **tuning.default**, **tuning.development**, **tuning.scientific**

PSSP: Installation and Migration Guide

Examples

1. To copy the **/tmp/my-tuning-file** file to the **/tftpboot/tuning.cust** file, enter:

```
cptuning /tmp/my-tuning-file
```

2. To copy the **/usr/lpp/ssp/install/config/tuning.commercial** file to the **/tftpboot/tuning.cust** file, enter:

```
cptuning tuning.commercial
```

create_dcehostname

Purpose

create_dcehostname – Populates the System Data Repository (SDR) with DCE hostnames for each node in a partition set to use DCE.

Syntax

create_dcehostname [–**h**] [–**v**]

Flags

–**h** Prints command syntax to standard output.

–**v** Prints out progress messages to standard output.

Operands

None.

Description

The **create_dcehostname** command must be run on the control workstation. It queries the DCE Security registry for information about nodes which may have already been configured and have current dcehostnames. For those entries found, it will update the SDR Node object's **dcehostname** attribute with this information. For all nodes which were not found in the DCE Security registry and do not already have a **dcehostname** attribute assigned, it will assign the node's reliable hostname to the attribute.Additionally, this program will update the SDR's SP object with the control workstation's dcehostname in the same manner it did for the nodes. All control workstation IP addresses will be used to search the DCE Security registry to determine if the control workstation has a defined DCE hostname in the registry. Since it is required that the control workstation be configured into the DCE cell, one of the IP addresses will be found in the registry.

Standard Input

CDS Registry and Security Server

SDR

Standard Output

Log file created: **/var/adm/SPlogs/auth_install/log**

Exit Values

0 Indicates successful completion of the command.

1 Indicates that an error occurred during the execution of the command. Review any reported errors either on the console or in the Log file.

The node boot process requires the DCE hostname. Authentication will not be properly set up, if at all, without this information and the node may not be accessible by some services or users.

Security

You must have root privilege and write access to the SDR for all partitions to run this command.

Location

/usr/lpp/ssp/bin/create_dcehostname

Related Information

DCE Administration Publications for AIX.

Examples

To create a DCE hostname for all defined nodes in the SDR, enter:

```
create_dcehostname -v
```

create_keyfiles

Purpose

create_keyfiles – Creates DCE keytab objects and stores them into specified keyfiles on the local file system. Services which use DCE as an authentication method will use these keys to log into DCE.

Syntax

create_keyfiles [−**h**] [−**v**] [−**c** | −**p** *partition_name*]

Flags

−h Prints out syntax of command to standard output.

−v Prints progress messages to standard output (for debugging).

−c Creates keyfiles for only those principals specific to the control workstation. This option is required when running this command prior to node number being available in the ODM and the SDR is not available.

−p Creates keyfiles only for the *partition_name* principals.

Operands

None.

Description

The **create_keyfiles** command reads from two files, a default file (**/usr/lpp/ssp/config/spsec_defaults**) and an override file (**/spdata/sys1/spsec/spsec_overrides**). It will process these files and generate an effective service principal based on the attributes specified for each service and its location (node, control workstation, or non−SP workstation). It will create keytab objects based on these effective service principals and store the keys in keyfiles located in the **/spdata/sys1/keyfiles** directory.

Standard Input

/usr/lpp/ssp/config/spsec_defaults

/spdata/sys1/spsec/spsec_overrides

Standard Output

Log file created: **/var/adm/SPlogs/auth_install/log**.

Keyfiles located in **/spdata/sys1/keyfiles** (with subdirectories based on the service name).

Exit Values

0 Indicates successful completion of the command.

1 Indicates that errors occurred during the execution of the command. Review any reported errors either on the console or in the Log file.

Security

This command requires write access to the **/spdata/sys1** filesystem, and read access to the two configuration files specified in the description. You must also be **root** with default dce credentials.

Location

/usr/lpp/ssp/bin/create_keyfiles

Related Information

File: **/usr/lpp/ssp/config/spsec_defaults**

File: **/spdata/sys1/spsec/spsec_overrides**

Commands: **rm_spsec**

DCE Administration Publications for AIX (relating to keyfiles and keytab object creation)

DCE Commands Reference for AIX

Examples

1. To create keyfiles for all services designated to run on the local machine as root user, enter:

   ```
   create_keyfiles -v
   ```

2. To create control workstation only service principals and the default partition service principals (used during initial install when the SDR is not available) as root user, enter:

   ```
   create_keyfiles -v
   ```

3. To create keyfiles for partition *my_par* services, enter:

   ```
   create_keyfiles -v -p my_par
   ```

create_krb_files

Purpose

create_krb_files – Creates the necessary **krb_srvtab** and **tftp** access files on the Network Installation Management (NIM) master for Kerberos Version 4 authentication.

Syntax

create_krb_files [−**h**]

Flags

−**h** Displays usage information. If the command is issued with the −**h** flag, the syntax description is displayed to standard output and no other action is taken.

Operands

None.

Description

Use this command on a boot/install server (including the control workstation). On the server, it creates the Kerberos Version 4 **krb_srvtab** file for each boot/install client of that server and also updates the **/etc/tftpaccess.ctl** file on the server.

Standard Error

This command writes error messages (as necessary) to standard error.

Exit Values

0 Indicates the successful completion of the command.

−**1** Indicates that an error occurred.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Location

/usr/lpp/ssp/bin/create_krb_files

Related Information

Commands: **setup_server**

Examples

To create or update the **krb_srvtab** and **tftp** access files on a boot/install server, enter the following command on that server:

```
create_krb_files
```

createhsd

Purpose

createhsd – Creates one hashed shared disk that encompasses two or more virtual shared disks.

Syntax

createhsd –**n** {*node_list* | **ALL**} –**s** *size_in_MB*
–**g** *volume _group_name* –**t** *stripe_size_in_KB*
[–**T** *lp_size_in_MB*] [{ –**c** *vsd_per_node* | –**L**} [–**A**]]
[–**S**] [–**o cache | nocache**] [–**m** *mirror_cnt*]
[–**d** *hsd_name*] [–**l** *lv_name_prefix*] [–**k** *vsd_type*] [–**x**]

Flags

Note: Some examples shown in this list do not contain enough flags to be executable. They are shown in an incomplete form to illustrate specific flags.

–**n** *node_list*
Specifies the nodes on which you are creating virtual shared disks. The backup node cannot be the same as the primary node. For VSD the node list is:

```
[P/S] : hdisk_list1+hdisk_list2/
```

For CVSD the node list is:

```
[S1/S2] : hdisk_list1+hdisk_list2/
```

P specifies the primary server node for serially accessed shared disks, S specifies the backup (secondary) server node for serially accessed shared disks, and S1 and S2 specifies the server nodes for concurrently accessed shared disks. *hdisk_list1* is the list of local physical disks in the logical volume on the primary, and *hdisk_list1+hdisk_list2* is the list of local physical disks in the volume group on the primary, if you want to have more disks in the volume group than are needed for the logical volume. The sequence in which nodes are listed determines the names given to the virtual shared disks. For example:

```
createvsd -n 1,6,4 -v PRE
```

(with the *vsd_prefix* PRE) creates virtual shared disks PRE1n1 on node 1, PRE2n6 on node 6, and PRE3n4 on node 4.

To create a volume group that spans hdisk2, hdisk3, and hdisk4 on node 1, with a backup on node 3, enter:

```
createvsd -n 1/3:hdisk2,hdisk3,hdisk4/ -v DATA
```

This command creates:

- virtual shared disk DATA1n1 with logical volume lvDATA1n1 on a volume group with the global volume group name DATA1n1b3 on node 1, exported to node 3. The Logical Volume Manager (LVM) volume group name is DATA. The logical volumes span hdisk2, hdisk3, and hdisk4.

To create volume groups just like that one on nodes 1, 2, and 3 of a system with backup on nodes 4, 5, and 6 of the same system, enter:

```
createvsd -n 1/4:hdisk1,hdisk2,hdisk3/,2/5:hdisk5,hdisk6, \

hdisk7/,3/6:hdisk2,hdisk4,hdisk6/ -v DATA
```

This command is shown on two lines here, but you must enter it without any spaces between the items in *node_list*.

The command creates:

- virtual shared disk DATA1n1 with logical volume lvDATA1n1 on a volume group with the local volume group name DATA on node 1, exported to node 4. The global volume group name is DATAn1b4.
- virtual shared disk DATA2n2 with logical volume lvDATA2n2 on a volume group with the local volume group name DATA on node 2, exported to node 5. The global volume group name is DATAn2b5.
- virtual shared disk DATA3n3 with logical volume lvDATA3n3 on a volume group with the local volume group name DATA on node 3, exported to node 6. The global volume group name is DATAn3b6.

To create a virtual shared disk where the logical volume spans only two of the physical disks in the volume group, enter:

```
createvsd -n 1/3:hdisk1,hdisk2+hdisk3/ -v DATA
```

This command creates the virtual shared disk DATA1n1 with logical volume lvDATA1n1 spanning hdisk1 and hdisk2 in the volume group DATA, which includes hdisk1, hdisk2, and hdisk3. It exports the volume group DATA to node 3.

If a volume group is already created and the combined physical hdisk lists contain disks that are not needed for the logical volume, those hdisks are added to the volume group. If the volume group has not already been created, **createvsd** creates a volume group that spans *hdisk_list1+hdisk_list2*.

Backup nodes cannot use the same physical disk as the primary does to serve virtual shared disks.

ALL specifies that you are creating virtual shared disks on all nodes in the system or system partition. No backup nodes are assigned if you use this operand. The virtual shared disks will be created on all the physical disks attached to the nodes in *node_list* (you cannot specify which physical disks to use.)

−s Specifies the total usable size of the hashed shared disk in MB. Unless −**S** is specified, **createhsd** adds at least a stripe size to each virtual shared disk's size for each hashed shared disk.

−g Specifies the Logical Volume Manager (LVM) volume group name, or local volume group name. This name is concatenated with the node number to form the global volume group name (VSD_GVG). For example:

```
createhsd -n 6 -g VSDVG
```

creates a new volume group with the local AIX volume group name VSDVG and the virtual shared disk global volume group name VSDVGn6. The node number is added to the local volume group name to create a unique global volume group name within a system partition to avoid name conflicts with the name used for volume groups on other nodes. If a backup node exists, the global volume group name will be created by concatenating the backup node number as well as the primary node number to the local volume group name. For example:

```
createhsd -n 6/3/ -g VSDVG
```

creates VSDVGn6b3, where the primary node is node 6 and the backup node for this global volume group is node 3. The local AIX volume group name will still be VSDVG. You can specify a local volume group that already exists. You do not need to use the **–T** flag if you specify a volume group name that already exists.

–t Specifies the stripe size in kilobytes that a hashed shared disk will use. The stripe size must be a multiple of 4KB and less than or equal to 1GB.

–T Specifies the size of the physical partition in the Logical Volume Manager logical volume group and also the logical partition size (they will be the same) in megabytes. You must select a power of 2 in the range 2—256. The default is 4MB.

The Logical Volume Manager limits the number of physical partitions to 1016 per disk. If a disk is greater than 4 gigabytes in size, the physical partition size must be greater than 4MB to keep the number of partitions under the limit.

–c Specifies the number of virtual shared disks to be created on each node. If *number_of_vsds_per_node* is not specified, one virtual shared disk is created for each node specified on **createvsd**. If more than one virtual shared disk is to be created for each node, the names will be allocated cyclically. For example:

```
createhsd -n 1,6 -c 2 -d DATA
```

creates virtual shared disks DATA1n1 on node 1, DATA2n6 on node 6, DATA3n1 on node 1, and DATA4n6 on node 6 and uses them to make up the hashed shared disk DATA.

–L Allows you to create one virtual shared disk on each node without using sequential numbers for locally-accessed IBM Virtual Shared Disks.

–A Specifies that virtual shared disk names will be allocated to each node in turn. For example:

```
createhsd -n 1,6 -c 2 -A DATA
```

creates DATA1n1 and DATA2n1 on node 1, and DATA3n6 and DATA4n6 on node 6.

–S Specifies that the hashed shared disk overrides the default skip option and does not skip the first stripe to protect the first LVM Control Block (LVCB).

–o Specifies either the **cache** or **nocache** option for the underlying virtual shared disks. The default is **nocache**.

–m Specifies the LVM mirroring count. The mirroring count sets the number of physical partitions allocated to each logical partition. The range is from 1 to 3. If **–m** is not specified, the count is set to 1.

–d Specifies the name assigned to the created hashed shared disk. It is used as the virtual shared disk prefix name (the **–v** in **createvsd**). If a hashed shared disk name is not specified, a default name, *x*HsD is used, where x denotes a sequence number.

The command:

```
createhsd -n 1,2 -d DATA
```

creates two virtual shared disks, DATA1n1 and DATA2n2. These virtual shared disks make up one hashed shared disk named DATA.

–l Overrides the prefix *lvx* that is given by default to a logical volume by the **createvsd** command, where *x* is the virtual shared disk name prefix specified by *vsd_name_prefix* or the default (`vsd`). For example:

```
createhsd -n 1 -v DATA
```

creates one virtual shared disk on node 1 named DATA1n1 with an underlying logical volume lvDATA1n1. If the command

```
createhsd -n 1 -v DATA -l new
```

is used, the virtual shared disk on node 1 is still named DATA1n1, but the underlying logical volume is named lvnew1n1.

It is usually more helpful **not** to specify **–l**, so that your lists of virtual shared disk names and logical volume names are easy to associate with each other and you avoid naming conflicts.

–k *vsd_type*
Specifies the type of virtual shared disk. The options are:

- **VSD**: specifies a normal serial access shared disk, or
- **CVSD**: specifies a concurrent access shared disk.

The default is **VSD**.

–x Specifies that the steps required to synchronize the underlying virtual shared disks on the primary and secondary nodes should **not** be performed; that is, the sequence:

- **varyoffvg** on the primary node
- **exportvg** on the primary node
- **importvg** on the secondary node
- **chvg** on the secondary node
- **varyoffvg** on the secondary node
- **varyonvg** on the primary nodes

is not done as part of the **createvsd** processing that underlies the **createhsd** command. This speeds the operation of the command and avoids unnecessary processing in the case where several IBM Virtual Shared Disks are being created on the same primary/secondary nodes. In that case, however, you should either **not** specify **–x** on the last **createhsd** in the sequence or issue the volume group commands listed above explicitly.

Operands

None.

Description

This command uses the sysctl facility.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit createhsd_dialog
```

or

```
smit vsd_data
```

and Select the Create an HSD option with the `vsd_data` fastpath.

Standard Output

For the following command:

```
createhsd -n 1/:hdisk2,hdisk3/ -g twinVG -s 1600 -t 8 -S -l \

twinLV -d twinHSD -c 4
```

The messages returned to standard output are:

```
OK:0:vsdvg -g twinVGn1 twinVG 1

OK:0:defvsd twinLV1n1 twinVGn1 twinHSD1n1 nocache

OK:0:defvsd twinLV2n1 twinVGn1 twinHSD2n1 nocache

OK:0:defvsd twinLV3n1 twinVGn1 twinHSD3n1 nocache

OK:0:defvsd twinLV4n1 twinVGn1 twinHSD4n1 nocache


OK:createvsd { -n 1/:hdisk2,hdisk3/ -s 401 -T 4 -g twinVG

-c 4 -v twinHSD -l twinLV -o cache -K }


OK:0:defhsd twinHSD not_protect_lvcb 8192 twinHSD1n1 twinHSD2n1

twinHSD3n1 twinHSD4n1
```

Exit Values

0 Indicates the successful completion of the command.

–1 Indicates that an error occurred.

Security

You must have access to the virtual shared disk subsystem via the sysctl service to run this command.

Restrictions

1. The backup node cannot be the same as the primary node.
2. The last character of *hsd_name* cannot be numeric.
3. The *vsd_name_prefix* cannot contain the character '**.**'. See the **createvsd -v** option for details.

Prerequisite Information

PSSP: Managing Shared Disks

Location

/usr/lpp/csd/bin/createhsd

Related Information

Commands: **createvsd**, **defhsd**, **vsdvg**

Examples

To create six 4MB virtual shared disks and their underlying logical volumes with a prefix of TEMP, as well as a hashed shared disk comprising those virtual shared disks (24MB overall) with a stripe size of 32KB, enter the following (assuming that no previous virtual shared disks are defined with the TEMP prefix):

```
createhsd -n 3,4,7/8/ -c 2 -s 1024 -g vsdvg -d TEMP -t 32
```

This creates the following virtual shared disks:

- TEMP1n3 with a logical volume lvTEMP1n3 defined in Logical Volume Manager volume group vsdvg3 of node 3.
- TEMP2n4 with a logical volume lvTEMP2n4 defined in volume group vsdvg4 of node 4.
- TEMP3n7 with a logical volume lvTEMP3n7 defined in volume group vsdvg7b8 of node 7 and imported to node8
- TEMP4n3 with a logical volume lvTEMP4n3 defined in volume group vsdvg3 of node 3
- TEMP5n4 with a logical volume lvTEMP5n4 defined in volume group vsdvg4 of node 4
- TEMP6n7 virtual shared disk with a logical volume lvTEMP6n7 defined in volume group vsdvg7 of node 7 and imported to node8

and the HSD:

- TEMP with a stripe size of 32KB that includes the preceding virtual shared disks in the order listed.

Note: TEMP does not write to the first 32KB of each of its virtual shared disks.

createvsd

Purpose

createvsd – Creates a set of virtual shared disks, with their associated logical volumes, and puts information about them into the System Data Repository (SDR).

Syntax

Note: Some examples shown in this list do not contain enough flags to be executable. They are shown in an incomplete form to illustrate specific flags.

createvsd –**n** {*node_list* | **ALL**} –**s** *size_in_MB* –**g** *vg_name*
[{–**c** *vsds_per_node* | –**L**}] [–A]]
[{–**m** *mirror_count* | –**p** *lvm_stripe_size_in_K*}] [–**v** *vsd_name_prefix*]
[–**l** *lv_name_prefix*] [–**o** **cache** | **nocache**]
[–**T** *lp_size_in_MB*] [–**k** *vsd_type*] [–**x**]

Flags

–**n** *node_list*
Specifies the nodes on which you are creating virtual shared disks. The backup node cannot be the same as the primary node. For VSD the node list is:

```
[P/S] : hdisk_list1+hdisk_list2/
```

For CVSD the node list is:

```
[S1/S2] : hdisk_list1+hdisk_list2/
```

P specifies the primary server node for serially accessed shared disks, S specifies the backup (secondary) server node for serially accessed shared disks, and S1 and S2 specifies the server nodes for concurrently accessed shared disks. *hdisk_list1* is the list of local physical disks in the logical volume on the primary, and *hdisk_list1+hdisk_list2* is the list of local physical disks in the volume group on the primary, if you want to have more disks in the volume group than are needed for the logical volume. The sequence in which nodes are listed determines the names given to the virtual shared disks. For example:

```
createvsd -n 1,6,4 -v PRE
```

(with the *vsd_prefix* PRE) creates virtual shared disks PRE1n1 on node 1, PRE2n6 on node 6, and PRE3n4 on node 4.

To create a volume group that spans hdisk2, hdisk3, and hdisk4 on node 1, with a backup on node 3, enter:

```
createvsd -n 1/3:hdisk2,hdisk3,hdisk4/ -v DATA
```

This command creates:

- virtual shared disk DATA1n1 with logical volume lvDATA1n1 on a volume group with the global volume group name DATA1n1b3 on node 1, exported to node 3. The Logical Volume Manager (LVM) volume group name is DATA. The logical volumes span hdisk2, hdisk3, and hdisk4.

To create volume groups just like that one on nodes 1, 2, and 3 of a system with backup on nodes 4, 5, and 6 of the same system, enter:

```
createvsd -n 1/4:hdisk1,hdisk2,hdisk3/,2/5:hdisk5,hdisk6, \

hdisk7/,3/6:hdisk2,hdisk4,hdisk6/ -v DATA
```

This command is shown on two lines here, but you must enter it without any spaces between the items in *node_list*.

The command creates:

- virtual shared disk DATA1n1 with logical volume lvDATA1n1 on a volume group with the local volume group name DATA on node 1, exported to node 4. The global volume group name is DATAn1b4.
- virtual shared disk DATA2n2 with logical volume lvDATA2n2 on a volume group with the local volume group name DATA on node 2, exported to node 5. The global volume group name is DATAn2b5.
- virtual shared disk DATA3n3 with logical volume lvDATA3n3 on a volume group with the local volume group name DATA on node 3, exported to node 6. The global volume group name is DATAn3b6.

To create a virtual shared disk where the logical volume spans only two of the physical disks in the volume group, enter:

```
createvsd -n 1/3:hdisk1,hdisk2+hdisk3/ -v DATA
```

This command creates the virtual shared disk DATA1n1 with logical volume lvDATA1n1 spanning hdisk1 and hdisk2 in the volume group DATA, which includes hdisk1, hdisk2, and hdisk3. It exports the volume group DATA to node 3.

If a volume group is already created and the combined physical hdisk lists contain disks that are not needed for the logical volume, those hdisks are added to the volume group. If the volume group has not already been created, **createvsd** creates a volume group that spans *hdisk_list1+hdisk_list2*.

Backup nodes cannot use the same physical disk as the primary does to serve virtual shared disks.

ALL specifies that you are creating virtual shared disks on all nodes in the system or system partition. No backup nodes are assigned if you use this operand. The virtual shared disks will be created on all the physical disks attached to the nodes in *node_list* (you cannot specify which physical disks to use.)

–s Specifies the size in megabytes of each virtual shared disk.

–g Specifies the Logical Volume Manager (LVM) volume group name. This name is concatenated with the node number to produce the global volume group name. For example:

```
createvsd -n 6 -g VSDVG
```

creates a volume group with the local volume group name VSDVG and the global volume group name VSDVG1n6 on node 6. The node number is added to the prefix to avoid name conflicts when a backup node takes over a volume group. If a backup node exists, the global volume group name will be concatenated with the backup node number as well as the primary. For example:

```
createvsd -n 6/3/ -g VSDVG
```

creates a volume group with the local volume group name VSDVG and the global volume group name VSDVGn6b3. The primary node is node 6 and the backup node for this volume group is node 3.

–c Specifies the number of virtual shared disks to be created on each node. If *number_of_vsds_per_node* is not specified, one virtual shared disk is created for each node specified on **createvsd**. If more than one virtual shared disk is to be created for each node, the names will be allocated alternately. For example:

```
createvsd -n 1,6 -c 2 -v DATA
```

creates virtual shared disks DATA1n1 on node 1, DATA2n6 on node 6, DATA3n1 on node 1, and DATA4n6 on node 6.

–L Allows you to create one virtual shared disk on each node without using sequential numbers, for locally-accessed virtual shared disks.

–A Specifies that virtual shared disk names will be allocated to each node in turn, for example:

```
createvsd -n 1,6 -c 2 -A DATA
```

creates DATA1n1 and DATA2n1 on node 1, and DATA3n6 and DATA4n6 on node 6.

–m Specifies the LVM mirroring count. The mirroring count sets the number of physical partitions allocated to each logical partition. The range is from 1 to 3 and the default value is 1.

–p Specifies the LVM stripe size. If this flag is not specified, the logical volumes are not striped. To use striping, the node on which the virtual shared disks are defined must have more than one physical disk.

–v Specifies a prefix to be given to the names of the created virtual shared disks. This prefix will be concatenated with the virtual shared disk number, node number, and backup node number, if a backup disk is specified. For example, if the prefix PRE is given to a virtual shared disk created on node 1 and there are already two virtual shared disks with this prefix across the partition, the new virtual shared disk name will be PRE3n1. The name given to the underlying logical volume will be lvPRE3n1, unless the –**l** flag is used. The **createvsd** command continues to sequence virtual shared disk names from the last PRE-prefixed virtual shared disk.

If **-v** is not specified, the prefix `vsd` is used.

Note: The last character of the *vsd_name_prefix* cannot be a digit. Otherwise, the 11th virtual shared disk with the prefix PRE would have the same name as the first virtual shared disk with the prefix PRE1. Nor can the *vsd_name_prefix* contain the character **'.'** because **'.'** can be any character in regular expressions.

–l Overrides the prefix *lvx* that is given by default to a logical volume by the **createvsd** command, where *x* is the virtual shared disk name prefix specified by *vsd_name_prefix* or the default (`vsd`). For example:

```
createvsd -n 1 -v DATA
```

creates one virtual shared disk on node 1 named DATA1n1 with an underlying logical volume lvDATA1n1. If the command

```
createvsd -n 1 -v DATA -l new
```

is used, the virtual shared disk on node 1 is still named DATA1n1, but the underlying logical volume is named lvnew1n1.

It is usually more helpful **not** to specify –**l**, so that your lists of virtual shared disk names and logical volume names are easy to associate with each other and you avoid naming conflicts.

–**o** Specifies either the **cache** or the **nocache** option. The default is **nocache**.

–**T** Specifies the size of the physical partition in the Logical Volume Manager logical volume group and also the logical partition size (they will be the same) in megabytes. You must select a power of 2 in the range 2 – 256. The default is 4MB.

The Logical Volume Manager limits the number of physical partitions to 1016 per disk. If a disk is greater than 4 gigabytes in size, the physical partition size must be greater than 4MB to keep the number of partitions under the limit.

–**k** *vsd_type*
Specifies the type of virtual shared disk. The options are:

- **VSD**: specifies a normal serial access shared disk, or
- **CVSD**: specifies a concurrent access shared disk.

The default is **VSD**.

–**x** Specifies that the steps required to synchronize the virtual shared disks on the primary and secondary nodes should **not** be performed; that is, the sequence:

- **varyoffvg** on the primary node
- **exportvg** on the secondary node
- **importvg** on the secondary node
- **chvg** on the secondary node
- **varyoffvg** on the secondary node
- **varyonvg** on the primary nodes

is not done as part of the **createvsd** processing. This speeds the operation of the command and avoids unnecessary processing in the case where several IBM Virtual Shared Disks are being created on the same primary/secondary nodes. In this case, however, you should either **not** specify –**x** on the last **createvsd** in the sequence or issue the volume group commands listed above explicitly.

Operands

None.

Description

Use this command to create a volume group with the specified name (if one does not already exist) and creates a logical volume of size **s** within that volume group.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit vsd_data
```

and select the Create a virtual shared disk option.

Standard Output

For the following command:

```
createvsd -n 1/:hdisk1/ -g testvg -s 16 -T 8 -l lvtest -v test -c 4
```

The messages returned to standard output are:

```
OK:0:vsdvg -g testvgn1 testvg 1

OK:0:defvsd lvtest1n1 testvgn1 test1n1 nocache

OK:0:defvsd lvtest2n1 testvgn1 test2n1 nocache

OK:0:defvsd lvtest3n1 testvgn1 test3n1 nocache

OK:0:defvsd lvtest4n1 testvgn1 test4n1 nocache
```

For the following command:

```
createvsd -n 1/:hdisk1/ -g testvg -s 16 -T 8 -l lvtest -v test -c 4
```

The messages returned to standard output are:

```
OK:0:defvsd lvtest5n1 testvgn1 test5n1 nocache

OK:0:defvsd lvtest6n1 testvgn1 test6n1 nocache

OK:0:defvsd lvtest7n1 testvgn1 test7n1 nocache

OK:0:defvsd lvtest8n1 testvgn1 test8n1 nocache
```

Exit Values

0 Indicates the successful completion of the command.

–1 Indicates that an error occurred.

Security

You must have access to the virtual shared disk subsystem via the sysctl service to run this command.

Restrictions

1. The backup node cannot be the same as the primary node.
2. The last character of *vsd_name_prefix* cannot be numeric.
3. The *vsd_name_prefix* cannot contain the character '**.**'.

Prerequisite Information

PSSP: Managing Shared Disks

Location

/usr/lpp/csd/bin/createvsd

Related Information

Commands: **defvsd**, **vsdvg**

Examples

To create two 4MB virtual shared disks on each of three primary nodes, one of which has a backup, enter:

```
createvsd -n 3,4,7/8/ -c 2 -s 4 -g vsdvg -v TEMP
```

This command creates the following virtual shared disks:

- TEMP1n3, with logical volume lvTEMP1n3 on a volume group with the global volume group name vsdvgn3 on node 3
- TEMP2n4, with logical volume lvTEMP2n4 on a volume group with the global volume group name vsdvgn4 on node 4
- TEMP3n7, with logical volume lvTEMP3n7 on a volume group with the global volume group name vsdvgn7b8 on node 7, also imported to node 8
- TEMP4n3, with logical volume lvTEMP4n3 on a volume group with the global volume group name vsdvgn3 on node 3
- TEMP5n4, with logical volume lvTEMP5n4 on a volume group with the global volume group name vsdvgn4 on node 4
- TEMP6n7, with logical volume lvTEMP6n7 on a volume group with the global volume group name vsdvgn7b8 on node 7, also imported to node 8

To create three virtual shared disks, where the logical volume created on node 3 spans fewer disks than the volume group does, enter:

```
createvsd -n 3,4/:hdisk1,hdisk2+hdisk3/,7/8/ -s 4 -g datavg -v USER
```

This command creates:

- USER1n3, with logical volume lvUSER1n3 defined on a volume group with the global volume group name datavgn3 on node 3.
- USER2n4, with logical volume lvUSER2n4 defined on a volume group with the global volume group name datavgn4 on node 4. datavgn4 spans hdisk1, hdisk2, and hdisk3. lvUSER2n4 spans hdisk1 and hdisk2.
- USER3n7, with logical volume lvUSER3n7 defined on a volume group with the global volume group name datavgn7b8 on node 7, also imported to node 8.

- If no volume group was defined on nodes 3 and 7 before this **createvsd** command was issued, the volume groups datavgn3 and datavgn7b8 are created with one 4MB partition from a single physical disk.

crunacct

Purpose

crunacct – Runs on the **acct_master** node to produce daily summary accounting reports and to accumulate accounting data for the fiscal period using merged accounting data from each node.

Syntax

crunacct [**-r**] |
[**-r SETUP** | **DELNODEDATA** | **MERGETACCT** | **CMS** | **USEREXIT** | **CLEANUP**]

Flags

-r Specifies a restart of the crunacct process. The restart process begins at the state listed in the statefile found in the **/var/adm/caact** directory.

Operands

SETUP Copies the files produced by **nrunacct** on each node to the **acct_master** node. For each node named by the string *node*, these files:

/var/adm/acct/nite/lineuse*YYYYMMDD*
/var/adm/acct/nite/reboots*YYYYMMDD*
/var/adm/acct/nite/daytacct*YYYYMMDD*
/var/adm/acct/sum/daycms*YYYYMMDD*
/var/adm/acct/sum/loginlog*YYYYMMDD*

are copied to the **acct_master** node to the following files:
/var/adm/cacct/*node***/nite/lineuse***YYYYMMDD*
/var/adm/cacct/*node***/nite/reboots***YYYYMMDD*
/var/adm/cacct/*node***/nite/daytacct***YYYYMMDD*
/var/adm/cacct/*node***/sum/daycms***YYYYMMDD*
/var/adm/cacct/*node***/sum/loginlog***YYYYMMDD*

for all *YYYYMMDD* prior or equal to the *YYYYMMDD* being processed.

DELNODEDATA
Deletes files that have been copied to the **acct_master** node in the SETUP step, as well as the associated **/var/adm/acct/statefile** *YYYYMMDD* files.

MERGETACCT
Produces a daily total accounting file and merges this daily file into the total accounting file for the fiscal period, for each accounting class. If there are no defined accounting classes, the output of this step represents data for the entire SP system.

CMS Produces a daily command summary file and merges this daily file into the total command summary file for the fiscal period, for each accounting class. If there are no defined accounting classes, the output of this step represents data for the entire SP system.

It also creates an SP system version of the **loginlog** file, in which each line consists of a date, a user login name and a list of node names. The date is the date of the last accounting cycle during which the user, indicated by the associated login name, had at least one connect session in the SP system. The associated list of node names indicates the nodes on which the user had a login session during that accounting cycle.

USEREXIT If the **/var/adm/csiteacct** shell file exists, calls it to perform site specific accounting procedures that are applicable to the **acct_master** node.

CLEANUP Prints a daily report of accounting activity and removes files that are no longer needed.

Description

In order for SP accounting to succeed each day, the **nrunacct** command must complete successfully on each node for which accounting is enabled and then the **crunacct** command must complete successfully on the **acct_master** node. However, this may not always be true. In particular, the following scenarios must be taken into account:

1. The **nrunacct** command does not complete successfully on some nodes for the current accounting cycle. This can be the result of an error during the execution of **nrunacct**, **nrunacct** not being executed at the proper time by **cron** or the node being down when **nrunacct** was scheduled to run.
2. The **acct_master** node is down or the **crunacct** command cannot be executed.

From the point of view of the **crunacct** command, the first scenario results in no accounting data being available from a node. The second scenario results in more than one day's accounting data being available from a node. If it is the case that no accounting data is available from a node, the policy of **crunacct** is that the error condition is reported and processing continues with data from the other nodes. If data cannot be obtained from at least X percent of nodes, then processing is terminated. "X" is referred to as the **spacct_actnode_thresh** attribute and can be set via a SMIT panel.

If node data for accounting cycle N is not available when **crunacct** executes and then becomes available to **crunacct** during accounting cycle N+1, the node data for both the N and N+1 accounting cycles is merged by **crunacct**. In general, **crunacct** merges all data from a node that has not yet been reported into the current accounting cycle, except as in the following case.

If it is the case that **crunacct** has not run for more than one accounting cycle, such that there are several day's data on each node, then the policy of **crunacct** is that it processes each accounting cycle's data to produce the normal output for each accounting cycle. For example, if **crunacct** has not executed for accounting cycles N and N+1, and it is now accounting cycle N+2, then **crunacct** first executes for accounting cycle N, then executes for accounting cycle N+1 and finally executes for accounting cycle N+2.

However, if the several accounting cycles span from the previous fiscal period to the current fiscal period, then only the accounting cycles that are part of the previous fiscal period are processed. The accounting cycles that are part of the current fiscal period are processed during the next night's execution of **crunacct**.

Appropriate messages are provided in the **/var/adm/cacct/active** file so that the administrator can execute **cmonacct** prior to the next night's execution of **crunacct**.

To restart the **crunacct** command after an error, first check the **/var/adm/cacct/active***YYYYMMDD* file for diagnostic messages, and take appropriate actions. For example, if the log indicates that data was unavailable from a majority of nodes, and their corresponding **nrunacct** state file indicate a state other than complete, check their **/var/adm/acct/nite/active***YYYYMMDD* files for diagnostic messages and then fix any damaged data files, such as **pacct** or **wtmp**.

Remove the lock files and lastdate file (all in the **/var/adm/cacct** directory), before restarting the **crunacct** command. You must specify the **-r** flag. The command begins processing cycles starting with the cycle after the last successfully completed cycle. This cycle will be restarted at the state specified in the statefile file. All subsequent cycles, up to and including the current cycle, will be run from the beginning (**SETUP** state).

You may choose to start the process at a different state by specifying a state with the **-r** flag. The command begins processing cycles starting with the cycle after the last successfully completed cycle. This cycle will be restarted at the state entered on the command line. All subsequent cycles, up to and including the current cycle, will be run from the beginning (**SETUP** state).

Files

/var/adm/cacct/activeYYYYMMDD
The **crunacct** message file.

/var/adm/cacct/fiscal_periods
Customer-defined file indicating start date of each fiscal period.

/var/adm/cacct/lastcycle
Contains last successful **crunacct** completed cycle.

/var/adm/cacct/lock*
Prevents simultaneous invocation of **crunacct**.

/var/adm/cacct/lastdate
Contains last date **crunacct** was run.

/var/adm/cacct/nite/statefileYYYYMMDD
Contains current state to process.

Security

You must have root privilege to run this command.

Prerequisite Information

For more information about the Accounting System, the preparation of daily and monthly reports, and the accounting files, see *PSSP: Administration Guide*.

Location

/usr/lpp/ssp/bin/crunacct

Related Information

Commands: **acctcms**, **acctcom**, **acctcon1**, **acctcon2**, **acctmerg**, **acctprc1**, **acctprc2**, **accton**, **crontab**, **fwtmp**, **nrunacct**

Daemon: **cron**

The System Accounting information found in *AIX Version 4.1 System Management Guide*

Examples

1. To restart the SP system accounting procedures, enter a command similar to the following:

   ```
   nohup /usr/lpp/ssp/bin/crunacct -r 2>> \

     /var/adm/cacct/nite/accterr &
   ```

   This example restarts **crunacct** at the state located in the statefile file. The **crunacct** command runs in the background (&), ignoring all INTERRUPT and QUIT signals (**nohup**). Standard error output (2) is added to the end (>>) of the **/var/adm/cacct/nite/accterr** file.

2. To restart the SP system accounting procedures at a specific state, enter a command similar to the following:

   ```
   nohup /usr/lpp/ssp/bin/crunacct -r CMS 2>> \

   /var/adm/cacct/nite/accterr &
   ```

   This example restarts the **crunacct** command starting with the CMS state. The **crunacct** command runs in the background (&), ignoring all INTERRUPT and QUIT signals (**nohup**). Standard error output (2) is added to the end (>>) of the **/var/adm/cacct/nite/accterr** file.

cshutdown

Purpose

cshutdown – Specifies the SP system **Shutdown** command.

Syntax

cshutdown [−**G**] [−**P**] [−**N** | −**g**] [−**R**] [−**W** *seconds*] [−**X** | −**E**] [−**Y**] [−**F**]
[−**h** | −**k** | −**m** | −**r** [−**C** *cstartup_options*]] [−**s**]
[{−**T** *minutes* | −**T** *hh:mm*} [−**M** *message_string*]]
[−**K** *number*] *target_nodes*

Flags

−**G** Allows the specification of nodes to include one or more nodes outside the current system partition. If **ALL** is specified with −**G**, all nodes in the SP are shut down. If **ALL** is specified without −**G**, all nodes in the current system partition are shut down. If −**G** is specified with a list of nodes, all listed nodes are shut down regardless of the system partition in which they reside (subject to the restrictions of the sequence file). If −**G** is not specified and some of the specified target nodes are outside of the current system partition or some of the specified target nodes depend on nodes outside of the current system partition, none of the specified nodes are shut down.

−**P** Powers off the nodes after the shutdown command completes. This is the default action except when the −**m** option (single user mode) is chosen.

−**N** Indicates that the *target_nodes* are specified as node numbers, not en0 host names. The node numbers can be specified as ranges, for example, 3-7 indicates nodes 3, 4, 5, 6, and 7.

−**g** Indicates that the *target_nodes* are specified as a named node group. If −**G** is supplied, a global node group is used. Otherwise, a partitioned-bound node group is used.

−**R** Indicates that *target_nodes* is a file that contains host identifiers. If you also use the −**N** flag, the file contains node numbers; otherwise, the file contains node names, specified as en0 host names.

−**W** *seconds*
Provides a time-out value for shutting down a leading node. In normal processing, **cshutdown** waits for a leading node to be completely halted before starting to shut down trailing nodes. If one or more leading nodes does not shut down, the **cshutdown** command waits indefinitely. The −**W** flag tells **cshutdown** to wait only the specified number of *seconds* after starting to halt a leading node; after that time, **cshutdown** starts the halt process for the trailing nodes.

Notes:

1. Be careful to use time-out values large enough to allow a node to complete shutdown processing. Your time-out value should be at least several minutes long; shorter values may be transparently modified to a higher value.
2. If shutdown processing for a node does not complete within the time-out

limit and **cshutdown** halts trailing nodes, the system may not function correctly.

If there are special subsystems, the same waiting procedure applies to subsystem sequencing in the subsystem phase.

−X Tells **cshutdown** that the state of nontarget nodes should not affect the result of the command. Use the **−X** flag to force **cshutdown** to shut down the target nodes if nontarget nodes listed in **/etc/cshutSeq** are gating the shutdown.

Note: If some critical nodes, but not the entire system, are forced to halt or reboot, the system may not function correctly.

−E Terminates processing if any nodes are found that are powered on, but not running (host_responds in the System Data Repository (SDR) shows a value of 0 − node shows red for hostResponds in SP Perspectives). This includes nodes that may have been placed in maintenance (single-user) mode. Refer to the “Description” section for additional information.

If you specify **−E**, you cannot specify **−X**.

−Y Tells **cshutdown** to ignore any error codes from the special subsystem interfaces. Without this flag, if a special subsystem interface exits with an error code, you receive a prompt allowing you to continue the operation, to quit, or to enter a subshell to investigate the error. On return from the subshell, you are prompted with the same choices.

−F Tells the **cshutdown** command to start the shut down immediately, without issuing warning messages to users.

−h Halts the target nodes. This is the default, unless overridden by the **−k**, **−m**, or **−r** flags.

−k Verifies the shutdown sequence file without shutting any node down. Special subsystems are not affected. There is no effect on a nonrunning target node. You can use **cshutdown −kF ALL** to test your **/etc/cshutSeq** file without actually shutting down any nodes and without sending messages to users.

−m Handles the request similar to a halt except that the last step, after syncing and unmounting file systems, is to bring the node to single user mode. There is no effect on a nonrunning target node.

−r Handles the request as a reboot. It performs the same operations as **−h**. Then it restarts the target nodes with **cstartup**. It does not power on a target node that was powered off at the time the **cshutdown** command was issued (it differs from the **cstartup** command, which powers on all specified nodes).

−C *cstartup_options*
Tells **cshutdown** to pass the *cstartup_options* to **cstartup** when the **cstartup** command is invoked after the *target_nodes* are halted. This flag is valid only when the **−r** (reboot) option is also specified. Any blanks in *cstartup_options* must be escaped or quoted.

−s Stops nonroot processes in the node order specified in **/etc/cshutSeq**. The default is to stop the nonroot processes in parallel.

[**−T** *time* [**−M** *message_string*]]
The **−T** flag specifies a *time* to start **cshutdown**, either as a number of minutes from now (**−T** *minutes*) or at the time in 24−hour format (**−T**

hh:*mm*). If the −**T** flag is specified, then you can use −**M** *message_string* to specify a message for users on the target nodes. Any blanks in *message_string* must be escaped or quoted.

−**K** Limits the number of concurrent processes created to **rsh** to the nodes. This is relevant to large systems. The default value is 64.

Operands

target_nodes Designates the target nodes to be operated on. It is the operand of the command, and must be the last token on the command line. In the absence of the −**R**, −**N**, or −**g** flags, *target_nodes* are specified as host names on the en0 Ethernet. Use **ALL** to designate the entire system. You must identify one or more *target_nodes*.

Description

Use this command to halt or reboot the entire system or any number of nodes in the system. The SP **cshutdown** command is analogous to the workstation **shutdown** command. Refer to the **shutdown** man page for a description of the **shutdown** command. The **cshutdown** command always powers off the nodes except while in Maintenance mode.

Note: If you bring a node down to maintenance mode, you must ensure file system integrity before rebooting the node.

In this case, the **cshutdown** command, which runs from the control workstation, cannot **rsh** to the node to perform the node shutdown phase processing. This includes the synchronization of the file systems. Therefore, you should issue the **sync** command three times in succession from the node console before running the **cshutdown** command. This is especially important if any files were created while the node was in maintenance mode.

To determine which nodes may be affected, issue the **spmon** −**d** −**G** command and look for a combination of power **on** and host_responds **no**.

For an SP system with a switch, if the entire system is being shutdown, issue the **Equiesce** command before issuing the **cshutdown** command. If only a portion of the system is being shutdown, but the switch primary node and the switch primary backup node are among the nodes targeted, use the **Eprimary** command to select a new switch primary node, and then issue the **Estart** command before issuing the **cshutdown** command.

The **cshutdown** command has these advantages over using the **shutdown** command to shut down each node of an SP:

- **cshutdown** provides a single point of control.

  Using one **cshutdown** command on the control workstation, you can shut down all or selected nodes.

- The sequencing of node shutdown and reboot is automated.

  You can use the **/etc/cshutSeq** file to control the order in which nodes are shut down, or you can let the system determine the order based on System Data Repository information about **/usr** servers and clients.

- Special subsystems can be notified of impending node shutdown.

The **/etc/subsysSeq** file lists these special subsystems and describes any sequencing relationships between them.

Shutdown processing has these phases:

1. Notifying all users of the impending shutdown, executing the customized shutdown script (**/etc/cshut.clean**) if it exists on the target node, then terminating all nonroot processes on the target nodes. Nonroot processes are sent a SIGTERM followed, 30 seconds later, by a SIGKILL. This gives user processes that handle SIGTERM a chance to do whatever cleanup is necessary.
2. Invoking any special subsystems, so they can perform any necessary shutdown activities. This phase follows the sequencing rules in **/etc/subsysSeq**. See *PSSP: Administration Guide* for the format of the **/etc/subsysSeq** file.
3. Starting node phase shutdown. The node phase includes syncing and unmounting file systems and halting the nodes, following the sequencing rules in **/etc/cshutSeq**. See *PSSP: Administration Guide* for the format of the **/etc/cshutSeq** file.
4. Rebooting the system, if requested by the **−r** flag.

Results

The **cshutdown** command may be gated by a problem with some subsystems or nodes to complete shutdown. In this case, look in the file created: **/var/adm/SPlogs/cs/cshut.***MMDDhhmmss***.***pid*

MMDDhhmmss Time stamp.

pid The process ID of the **cshutdown** command.

If a file with the same name already exists (from a previous year), the **cshutdown** command overwrites the existing file.

Files

The following files reside on the control workstation:

/etc/cshutSeq Describes the sequence in which the nodes should be shut down. Nodes not listed in the file are shut down concurrently with listed nodes. If the file is empty, all nodes are shut down concurrently. If the file does not exist, **cshutdown** uses the output of **seqfile** as a temporary sequencing default.

/etc/subsysSeq
Describes groups of special subsystems that need to be invoked in the subsystem phase of **cshutdown**. Also shows the sequence of invocation. Subsystems are represented by their invocation commands. If this file does not exist or is empty, no subsystem invocation is performed.

/var/adm/SPlogs/cs/cshut.*MMDDhhmmss***.***pid*
Road map of **cshutdown** command progress.

The following file may reside on the target nodes:

/etc/cshut.clean
Name of the customized shutdown script that will be run before **cshutdown** terminates nonroot processes. This script is created by the user to stop nonroot processes gracefully before **cshutdown** terminates them.

Security

The **cshutdown** command can only be issued on the control workstation. To run the command you must have one of the following:

- root privilege
- membership in the AIX shutdown group
- membership in the AIX cshutgroup and access to the Hardware Monitor subsystem with VFOP (virtual front panel operator) permission on the hardware objects (frames, slots) being operated on.

Location

/usr/lpp/ssp/bin/cshutdown

Related Information

PSSP Commands: **cstartup**, **init**, **seqfile**, **shutdown**

AIX Commands: **rsh**

Examples

1. For these examples, assume that **/etc/cshutSeq** contains the following lines:

   Group1 > Group2 > Group3

   Group1: A

   Group2: B

   Group3: C

   This defines 3 groups, Group1 through Group3, each containing a single node. The nodes names are A, B, and C. The sequence line `Group1 > Group2 > Group3` means that Group3 (node C) is shut down first. When Group3 is down, Group2 (node B) is shut down. When Group2 is down, then Group1 (node A) is shut down.

   Table 1 on page 73 shows that the result of a **cshutdown** command depends on the flags specified on the command line, the initial state of each node, and the sequencing rules in **/etc/cshutSeq**. The shorthand notation A^{up} indicates that node A is up and running; A_{dn} indicates that node A is down.

Table 1. Examples of the cshutdown Command. The subscript ^{up} means the node is powered up and running; the subscript $_{dn}$ means the node is not running.

Initial State	Command Issued	Final State	Explanation
A^{up} B^{up} C^{up}	**cshutdown** A B C	A_{dn} B_{dn} C_{dn}	The command succeeds; the nodes are all down.
A^{up} B^{up} C_{dn}	**cshutdown** B	A^{up} B_{dn} C_{dn}	The command succeeds because C is already not running.
A^{up} B^{up} C_{dn}	**cshutdown** A	Unchanged	The command fails because B is still running.
A^{up} B^{up} C_{dn}	**cshutdown** −**X** A	A_{dn} B^{up} C_{dn}	The command succeeds because −**X** considers the sequencing of *only* the target nodes.

2. To shut down all the nodes in the SP system regardless of system partitions and the sequence file, enter:

   ```
   cshutdown -GXY ALL
   ```

3. To shut down nodes **1**, **9**, and **16—20** regardless of system partitions and subject to the restrictions of the sequence file, enter:

   ```
   cshutdown -G -N 1 9 16-20
   ```

   The command may be unsuccessful if any node in the list depends on any node that is not on the list and that node is not shutdown.

4. To shut down all the nodes in the current system partition, enter:

   ```
   cshutdown ALL
   ```

   The command may be unsuccessful if any node in the current system partition depends on nodes outside of the current system partition.

5. To shut down nodes **1**, **5**, and **6** in the current system partition, enter:

   ```
   cshutdown -N 1 5 6
   ```

   The command may be unsuccessful if any node in the list is not in the current system partition or depends on nodes outside of the current system partition.

6. Specify the −**X** flag to ignore the sequence file and force nodes **1**, **5**, and **6** to be shut down. The following command is successful even if node 5 is gated by a node that is not shut down or is outside the current system partition:

   ```
   cshutdown -X -N 1 5 6
   ```

7. To do a fast shut down on node 5 without sending a warning message to the user, enter:

   ```
   cshutdown -F -N 5
   ```

8. To verify the sequence file without shutting down any node, enter the −**k** flag as follows. If both the −**k** and −**F** flags are specified, the sequence file can be tested without actually shutting down any nodes and without issuing a warning message to the user.

   ```
   cshutdown -kF ALL
   ```

9. Specify the −**r** flag to halt the target nodes and restart them with **cstartup**. If necessary, specify the −**C** flag to provide *cstartup_options*. For example, to halt

and restart nodes 12—16 with a time-out value of 300 seconds for the purpose of starting a leading node, enter:

```
cshutdown -rN -C'-W 300' 12-16
```

10. To reboot all the nodes in the partition node group sleepy_nodes, enter:

```
cshutdown -rg sleepy_nodes
```

CSS_test

Purpose

CSS_test – Verifies that the installation and configuration of the Communications Subsystem of the SP system completed successfully.

Syntax

CSS_test

Flags

None.

Operands

None.

Description

Use this command to verify that the Communications Subsystem component **ssp.css** of the SP system was correctly installed. **CSS_test** runs on the system partition set in SP_NAME.

A return code of 0 indicates that the test completed without an error, but unexpected results may be noted on standard output and in the companion log file **/var/adm/SPlogs/CSS_test.log**. A return code of 1 indicates that an error occurred.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit SP_verify
```

Files

/var/adm/SPlogs/CSS_test.log
Default log file

Location

/usr/lpp/ssp/bin/CSS_test

Related Information

Commands: **st_verify**, **SDR_test**, **SYSMAN_test**, **spmon_ctest**, **spmon_itest**

Examples

To verify the Communication Subsystem following installation, enter:

```
CSS_test
```

css.snap

Purpose

css.snap – Collects switch related log and trace files from a node.

Syntax

css.snap [−**c** | −**n** | −**s**]

Flags

−**c** Erases the contents of the adapter cache and prints the result (Default).

−**n** Assumes the device driver or daemon has erased the contents of the cache.

−**s** Soft snap will exclude the **tb_dump** log. Used for temporary (TYPE=TEMP) error where the viability of the adapter is in doubt, or it is not desirable to damage the adapter state by the use of diagnostic routines.

Operands

None.

Description

css.snap is generally issued automatically from the **fault_service** daemon when switch related errors occur and the data may be of use in debugging a problem. It can also be issued by the system administrator, usually under the direction of IBM level 2 or PE support. **css.snap** can be run on nodes with SP switch adapters or on the control workstation. It always collects logs local to the node from which it is run.

Files

/var/adm/SPlogs/css/css.snap.log
Specifies the trace file.

/var/adm/SPlogs/css/hostname.dateymdHMS.css.snap.tar.Z
Specifies the compressed tar file containing switch logs and debug information.

Location

/usr/lpp/ssp/css/css.snap

Examples

To collect data because Estart was unsuccessful on the switch Primary node (c191n01) enter:

```
ﬄc191n01“> /usr/lpp/ssp/css/css.snap
```

cstartup

Purpose

cstartup – Specifies the SP system **Startup** command.

Caution!

The **cstartup** command attempts to power on nodes that are powered off. This has safety implications if someone is working on the nodes. Proper precautions should be taken when using this command.

Syntax

cstartup [−**E**] [−**G**] [−**k**] [−**N** | −**R** | −**g**] [−**S**] [−**W** *seconds*] [−**X**]
[−**Z**] [−**z**] { *target_nodes* | [**ALL**]}

Flags

−E Starts up all nodes concurrently. Ignores the **/etc/cstartSeq** file, if one exists.

−G Allows the specification of nodes to include one or more nodes outside of the current system partition. If **ALL** is specified with −**G**, all nodes in the SP start up. If **ALL** is specified without −**G**, all nodes in the current system partition start up. If −**G** is specified with a list of nodes, all listed nodes start up regardless of the system partition in which they reside (subject to the restrictions of the sequence file). If −**G** is not specified and some of the specified target nodes are outside of the current system partition or some of the specified target nodes depend on nodes outside of the current system partition, none of the specified nodes are started up.

−g Indicates that the *target_nodes* are specified as a named node group. If −**G** is supplied, a global node group is used. Otherwise, a partitioned-bound node group is used.

−k Checks the sequence data file; does not start up any nodes. If circular sequencing is detected, **cstartup** issues warning messages. You can use **cstartup** −**k ALL** to test your **/etc/cstartSeq** file without starting or resetting any nodes.

−N Indicates that the *target_nodes* are specified as node numbers, not en0 host names. The node numbers can be specified as ranges; for example, 3−7 is interpreted as nodes 3, 4, 5, 6, and 7.

−R Indicates that *target_nodes* is a file that contains the node identifiers.

−S Tells **cstartup** to ignore existing sequencing violations; some trailing *target_nodes* are already up and running. The *target_nodes* that are already up are left alone. The other *target_nodes* are started in sequence. This operation may cause the nodes involved to not interface properly with their dependent nodes. If you omit the −**S** flag and any *target_node* is already running before its leading node, **cstartup** encounters an error without modifying the state of the system.

−W *seconds*
Provides a timeout value for starting up a leading node. In normal processing, **cstartup** waits for a leading node to be completely started

before initiating the startup of trailing nodes. If one or more *target_nodes* does not come up, **cstartup** waits indefinitely. The −**W** flag tells **cstartup** to wait the specified amount of time after initiating the startup of a node; the command continues to start other nodes, preserving the sequence in **/etc/cstartSeq**. The value you specify as *seconds* is added to a 3 minute (180 second) default wait period. Your value is a minimum; internal processing may cause the actual wait time to be slightly longer.

Note: Your system may still be usable if one or more nodes does not complete startup, because the sequencing rules are preserved.

−**X** Starts up only the nodes listed on the command line even if there are nontarget nodes gating the system startup. If you do not specify the −**X** flag and there are sequence violations involving nontarget nodes, **cstartup** encounters an error without modifying the state of the system.

Note: If some nodes but not the entire system are forced to start up this way, they may not function properly because of possible resource problems.

−**Z** If a *target_node* is already running at the time the **cstartup** command is issued, this flag tells **cstartup** to reset the node. This operation is disruptive to any processes running on the node. If you omit the −**Z** flag and any *target_node* is already running, **cstartup** encounters an error without modifying the state of the system.

−**z** If a *target_node* is already running at the time the **cstartup** command is issued, this flag tells **cstartup** to reset the node if the node is dependent on a node that is down when **cstartup** is issued, but leave the node alone if the node is to be started up ahead of any down node. This operation is disruptive to any processes running on the node being reset. This operation correctly resets the node-startup sequencing with minimum disruption to the system. If you omit the −**z** flag and any *target_node* is already running, **cstartup** encounters an error without modifying the state of the system.

Operands

target_nodes Designates the target nodes to be operated on. It is the operand of the command, and must be the last token on the command line. In the absence of the −**R**, −**N**, or −**g** flags, *target_nodes* are specified as host names on the en0 Ethernet. The string **ALL** can be used to designate all nodes in the SP system. You must identify one or more *target_nodes*.

Description

The **cstartup** command starts up the entire system or any number of nodes in the system. If a node is not powered on, startup means powering on the node. If the node is already powered on and not running, startup means resetting the node.

The **/etc/cstartSeq** file specifies the sequence in which the nodes are started up. See *PSSP: Administration Guide* for the format of the **/etc/cstartSeq** file.

You can use the −**SXZ** flags to violate the **cstartup** sequence intentionally. See Table 2 on page 80 for examples of the effect of these flags.

Results

The **/var/adm/SPlogs/cs/cstart.***MMDDhhmmss***.** *pid* file contains the results of **cstartup**.

MMDDhhmmss The time stamp.

pid The process ID of the **cstartup** command.

If the command is unsuccessful, examine this file to see which steps were completed. If a file with the same name already exists (from a previous year), the **cstartup** command overwrites the existing file.

Files

The following files reside on the control workstation:

/etc/cstartSeq Describes the sequence in which the nodes should be started. Nodes not listed in the file are started up concurrently with listed nodes. If the file is empty, all nodes are started up concurrently. If the file does not exist, **cstartup** uses the output of **seqfile** as a temporary sequencing default.

/var/adm/SPlogs/cs/cstart.*MMDDhhmmss***.***pid*
Road map of **cstartup** command progress.

Security

The **cstartup** command can only be issued on the control workstation. To run the command you must have one of the following:

- root privilege
- membership in the AIX shutdown group
- membership in the AIX cshutgroup and access to the Hardware Monitor subsystem with VFOP (virtual front panel operator) permission on the hardware objects (frames, slots) being operated on.

Location

/usr/lpp/ssp/bin/cstartup

Related Information

PSSP Commands: **cshutdown**, **init**, **seqfile**

AIX Commands: **rsh**

Examples

1. For these examples, assume that **/etc/cstartSeq** specifies the following startup sequence:

   Group1 > Group2 > Group3 > Group4 > Group5

   Group1: A

   Group2: B

   Group3: C

   Group4: D

Group5: E

This defines five groups, Group1 through Group5, each containing a single node. The nodes names are A, B, C, D, and E. The sequence line `Group1 > Group2 > Group3 > Group4 > Group5` means that Group1 (node A) is started first. When Group1 is up, Group2 (node B) is started. When Group2 is up, then Group3 (node C) is started, and so on.

Table 2 shows that the result of a **cstartup** command depends on the flags specified on the command line, the initial state of each node, and the sequencing rules in **/etc/cstartSeq**. The shorthand notation A^{up} indicates that A is powered up and running; A_{dn} indicates that A is not running.

Table 2 (Page 1 of 2). Examples of the cstartup Command. The subscript ^{up} means the node is up; the subscript $_{dn}$ means the node is down.

Initial State	Command Issued	Final State	Explanation
A_{dn} B_{dn} C_{dn} D_{dn} E_{dn}	**cstartup** A B C D E	A^{up} B^{up} C^{up} D^{up} E^{up}	The command succeeds; the nodes are all up.
A^{up} B^{up} C_{dn} D_{dn} E_{dn}	**cstartup** A B C D E	A^{up} B^{up} C^{up} D^{up} E^{up}	The command succeeds, C, D, and E are started up.
A^{up} B^{up} C_{dn} D^{up} E_{dn}	**cstartup** A B C D E	Unchanged	The command fails because D was already up before C.
A^{up} B^{up} C_{dn} D^{up} E_{dn}	**cstartup** −**S** A B C D E	A^{up} B^{up} C^{up} D^{up} E^{up}	The command succeeds because −**S** ignores sequencing violations.
A^{up} B^{up} C_{dn} D^{up} E_{dn}	**cstartup** −**Z** A B C D E	A^{up} B^{up} C^{up} D^{up} E^{up}	The command succeeds because −**Z** resets running nodes.
A^{up} B^{up} C_{dn} D^{up} E_{dn}	**cstartup** C E	Unchanged	The command fails because node D was already up before node C.
A^{up} B^{up} C_{dn} D^{up} E_{dn}	**cstartup** −**S** C E	A^{up} B^{up} C^{up} D^{up} E^{up}	The command succeeds because −**S** ignores sequencing violations.
A^{up} B^{up} C_{dn} D^{up} E_{dn}	**cstartup** −**X** C E	A^{up} B^{up} C^{up} D^{up} E^{up}	The command succeeds because −**X** considers the sequencing of *only* the target nodes.
A^{up} B^{up} C_{dn} D^{up} E_{dn}	**cstartup** −**Z** C E	unchanged	The command fails because resetting C or E does not correct the sequence violation.
A^{up} B^{up} C_{dn} D_{dn} E_{dn}	**cstartup** C E	unchanged	The command fails because D is gating E. Node C is not started either.

Table 2 (Page 2 of 2). Examples of the cstartup Command. The subscript $_{up}$ means the node is up; the subscript $_{dn}$ means the node is down.

Initial State	Command Issued	Final State	Explanation
A_{up} B_{up} C_{dn} D_{dn} E_{dn}	**cstartup** –**S** C E	unchanged	The command fails because D is gating E. Node C is not started either.
A_{up} B_{up} C_{dn} D_{dn} E_{dn}	**cstartup** –**X** C E	A_{up} B_{up} C_{up} D_{dn} E_{up}	The command succeeds and starts up only the explicit targets, C and E.
A_{up} B_{up} C_{dn} D_{dn} E_{dn}	**cstartup** –**Z** C E	unchanged	The command fails because D is gating E. Node C is not started either.

2. To start up all the nodes in the SP system regardless of system partitions and the sequence file, enter:

   ```
   cstartup -GXZ ALL
   ```

3. To start up nodes **1**, **9**, and **16—20** regardless of system partitions and subject to the restrictions of the sequence file, enter:

   ```
   cstartup -G -N 1 9 16-20
   ```

   The command may be unsuccessful if any node in the list depends on any node that is not on the list and that node is not started up.

4. To start up all the nodes in the current system partition, enter:

   ```
   cstartup ALL
   ```

   The command may be unsuccessful if any node in the current system partition depends on nodes outside of the current system partition.

5. To start up nodes **1**, **5**, and **6** in the current system partition, enter:

   ```
   cstartup -N 1 5 6
   ```

   The command may be unsuccessful if any node in the list is not in the current system partition or depends on nodes outside of the current system partition.

6. Specify the –**X** flag to ignore the sequence file and force nodes **1**, **5**, and **6** to be started up. The following command is successful even if node 5 is gated by a node that is not started up or is outside the current system partition:

   ```
   cstartup -X -N 1 5 6
   ```

7. To verify the sequence file without actually starting up or resetting any nodes, enter the –**k** flag as follows:

   ```
   cstartup -k ALL
   ```

8. To ignore the sequence file and start up all the target nodes concurrently, use the –**E** flag. For example, to start up all the nodes in the current system partition concurrently, enter:

   ```
   cstartup -E ALL
   ```

9. To start up all nodes in the system node group sleepy_nodes, enter:

```
cstartup -Gg sleepy_nodes
```

ctlhsd

Purpose

ctlhsd – Sets the operational parameters for the Hashed Shared Disk subsystem on a node.

Syntax

ctlhsd [−**p** *parallel_level* | −**v** *hsd_name* ... | −**C** | −**V**]

Flags

no option Displays the current parallelism level, the number of reworked requests, and the number of requests that are not at a page boundary.

−**p** *parallel_level*
Sets the HSD device driver's parallelism level as the specified value of the *parallel_level*.

−**v** *hsd_name* **...**
Resets the statistics in the number of reads and writes on the specified hashed shared disks.

−**C** Resets the HSD device drivers counters in the number of reworked requests and the number of read/write requests that are not at a page boundary.

−**V** Resets all the configured hashed shared disk's statistics in the number of read and write requests.

Operands

None.

Description

Use this command to set the parallelism level and to reset the statistics of the Hashed Shared Disk subsystem's data striping device driver for the virtual shared disk. When specified with no arguments, it displays the the current parallelism level, the number of reworked requests, and the number of requests that were not at a page boundary. When **ctlhsd** is used to reset the statistics of the device driver, or a particular hashed shared disk, or all the configured hashed shared disks on the system, it will not suspend all the underlying virtual shared disks. In other words, the user should make sure that there are no I/O activities on the underlying virtual shared disks.

Use **lshsd** −**s** to display the statistics on the number of read and write requests at the underlying virtual shared disks in a hashed shared disk or all hashed shared disks. Use the −**v** or −**V** flag to reset these counters.

Security

You must be in the AIX **bin** group to run this command.

Prerequisite Information

PSSP: Managing Shared Disks

Location

/usr/lpp/csd/bin/ctlhsd

Related Information

Commands: **cfghsd**, **lshsd**, **lsvsd**, **resumevsd**, **suspendvsd**, **ucfghsd**

Examples

To display the current parallelism level and counter, enter:

```
ctlhsd
```

The system displays a message similar to the following:

```
The current parallelism level is 9.

The number of READ requests not at page boundary is 0.

The number of WRITE requests not at page boundary is 0.
```

ctlvsd

Purpose

ctlvsd – Sets the operational parameters for the IBM Virtual Shared Disk subsystem on a node.

Syntax

ctlvsd [−**c** *cache_size* | −**r** *node_number*... | −**R** | −**p** *parallelism* | [−**l** **on** | **off**] −**k** *node_number*... | −**t** | −**T** | −**v** *vsd_name* ... | −**V** | −**C** | −**K** | −**M** *IP_max_message_size*]

Flags

−c Sets the cache size to the new value. Only increasing the cache size up to the maximum value is supported. The initial value of the cache size is the *init_cache_buffer_count* from the SDR Node object for the node.

−r Resets the outgoing and expected sequence numbers for the nodes specified on the node on which the command is run. Use this flag when another node has either been rebooted, cast out or all virtual shared disks have been reconfigured on that node. The specified nodes are also cast in.

−R Resets the outgoing and expected sequence number for all nodes on the node on which the command is run. Use this flag after rebooting the node. All nodes in the IBM Virtual Shared Disk network will be cast in.

−p Sets the level of IBM Virtual Shared Disk parallelism to the number specified. The valid range is 1 to 9. The default is 9. A larger value can potentially give better response time to large requests. (Refer to *PSSP: Managing Shared Disks* for more information regarding tuning IBM Virtual Shared Disk performance.)

This value is the *buf_cnt* parameter on the **uphysio** call that the IBM Virtual Shared Disk IP device driver makes in the kernel. Use **statvsd** to display the current value on the node on which the command is run.

−l on | off
Specify −**l on** to activate KLAPI. Specify −**l off** to deactivate KLAPI.

−k Casts out the node numbers specified on the local node. The local node ignores requests from cast out nodes. Use −**r** to cast nodes back in.

Note: Before using this flag, refer to the “Restrictions” section that follows.

−t Lists the current routing table and mbuf headers cached by the IBM Virtual Shared Disk driver.

−T Clears or releases all cached routes.

−v *vsd_name* **...**
Resets the statistics in the number of read and write requests on the specified virtual shared disks.

−V Resets all the configured virtual shared disk's statistics in the number of read and write requests.

−**C** Resets the IBM Virtual Shared Disk device driver counters displayed by the **statvsd** command. Exceptions are the outgoing and expected request sequence numbers among the client and server nodes.

−**K** Casts out all nodes on the local node. Local requests are still honored.

Note: Before using this flag, refer to the "Restrictions" section that follows.

−**M** Sets the IBM Virtual Shared Disk *max_IP_msg_size*. This is the largest sized block of data the virtual shared disk sends over the network for an I/O request. This limit also affects local virtual shared disk I/O block size. The value must be a multiple of 512 and between 512 and 65024 (64KB-512KB). IBM suggests using 65024 for the switch, and 24576 (24KB) for token-ring or Ethernet networks. (Refer to *PSSP: Managing Shared Disks* for more information regarding tuning IBM Virtual Shared Disk performance.) Use **statvsd** to display the current value on the node on which the command is run. Set to the same value on all nodes.

Operands

None.

Description

The **ctlvsd** command changes some parameters of the IBM Virtual Shared Disk subsystem. When called with no arguments it displays the current and maximum cache buffer count, the request block count, the pbuf count, the minimum buddy buffer size, the maximum buddy buffer size as well as the overall size of the buddy buffer.

Use **statvsd** to display outgoing and expected sequence numbers and out cast status of other nodes as viewed by the node on which the command is run. It is best to **suspendvsd** −**a** on all nodes whose sequence numbers are being reset prior to actually resetting the sequence numbers. Be sure to use **resumevsd** on all virtual shared disks that were suspended after resetting the sequence numbers.

Initially, all sequence numbers are set to 0 when the first virtual shared disk is configured and the IBM Virtual Shared Disk device driver is loaded. Thereafter, sequence numbers are incremented as requests are sent to (outgoing) and received from (expected) other nodes, and reset via **ctlvsd** −**R** | −**r** commands.

Reloading the IBM Virtual Shared Disk device driver by **suspendvsd** −**a**, **stopvsd** −**a**, or **ucfgvsd** −**a** followed by **cfgvsd** also resets all sequence numbers to 0.

Initially, all nodes in the IBM Virtual Shared Disk network are cast in. The **ctlvsd** −**k** command casts a node out. The local node ignores requests from cast out nodes. The **ctlvsd** −**r** command casts nodes back in.

Security

You must be in the AIX **bin** group to run this command.

Restrictions

If you have the IBM Recoverable Virtual Shared Disk software installed and operational, do not use the **-k** and **-K** options. The results may be unpredictable.

See *PSSP: Managing Shared Disks*.

Prerequisite Information

PSSP: Managing Shared Disks

Location

/usr/lpp/csd/bin/ctlvsd

Related Information

Commands: **cfgvsd**, **lsvsd**, **preparevsd**, **resumevsd**, **startvsd**, **statvsd**, **stopvsd**, **suspendvsd**, **ucfgvsd**

Refer to *PSSP: Managing Shared Disks* for information on tuning IBM Virtual Shared Disk performance and sequence numbers.

Examples

To display the current parameters, enter:

```
ctlvsd
```

The system displays a message similar to the following:

```
The current cache buffer count is 64.

The maximum cache buffer count is 256.

The request block count is 256.

The pbuf's count is 48.

The minimum buddy buffer size is 4096.

The maximum buddy buffer size is 65536.

The total buddy buffer size is 4 max buffers, 262144 bytes.
```

To display the mbuf headers and current routing table, enter:

```
ctlvsd -t
```

The system displays the following information:

```
Mbuf Cache Stats:

                Header

   Cached        1

      Hit     1023

     Miss        1

Route cache information:

 destination  interface  ref  status  direct/gateway   min managed mbuf

     1         css0        2    Up         Direct               256
```

defhsd

Purpose

defhsd – Designates a node as either having or using a hashed shared disk.

Syntax

defhsd *hsd_name* {**protect_lvcb** | **not_protect_lvcb**} *stripe_size* *vsd_name...*

Flags

None.

Operands

protect_lvcb | not_protect_lvcb
Protects the logical volume control block information that is stored at the first block of a logical volume. If **protect_lvcb** is specified, the data striping device will skip the first stripe on each underlying virtual shared disk in an HSD. In this case, you should define each logical volume one stripe larger than necessary. If the virtual shared disk software and Logical Volume Manager (LVM) disk mirroring are both used, the logical volume control block information is critical.

hsd_name Specifies a unique name for the new HSD. This name must be unique across the system partition and should be unique across the SP to avoid any naming conflicts during future system partitioning operations. The length of the name must be less than or equal to 31 characters.

stripe_size Specifies the maximum size of data stored on a virtual shared disk at one time. The smallest stripe size is 4096 bytes. The stripe size must be a multiple of 4096 and less than or equal to 1GB.

vsd_name Specifies the virtual shared disks that compose the HSD. All underlying virtual shared disks in the HSD must be defined before using this command.

Description

The **defhsd** command is used to specify the *hsd_name*, *stripe size* and underlying virtual shared disks for the new hashed shared disk.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit vsd_data
```

and select the Define a Hashed Shared Disk option.

Security

You must be in the AIX **bin** group and have write access to the SDR to run this command.

Prerequisite Information

PSSP: Managing Shared Disks

Location

/usr/lpp/csd/bin/defhsd

Related Information

Commands: **hsdatalst**, **undefhsd**, **updatehsd**

Refer to *PSSP: Managing Shared Disks* for information on tuning IBM Virtual Shared Disk performance and sequence numbers.

Examples

The following example adds SDR information indicating a stripe size of 32768, composed of *vsd.vsdn101*, *vsd.vsdn201*, and the name *hsd1* is defined.

```
defhsd hsd1 protect_lvcb 32768 vsd.vsdn101 vsd.vsdn201
```

defvsd

Purpose

defvsd – Designates a node as either having or using a virtual shared disk.

Syntax

defvsd *logical_volume_name global_group_name vsd_name* [**nocache** | **cache**]

Flags

None.

Operands

logical_volume_name
Is the name of the logical volume you want to specify as a virtual shared disk. This logical volume must reside on the global volume group indicated. The length of the name must be less than or equal to 15 characters.

global_group_name
Is the name of the globally-accessible volume group previously defined by the **vsdvg** command where you want to specify a virtual shared disk. The length of the name must be less than or equal to 31 characters.

vsd_name
Specifies a unique name for the new virtual shared disk. This name must be unique across the system partition and should be unique across the SP, to avoid any naming conflicts during future system partitioning operations. The suggested naming convention is **vsdnn***gvg_name*. The length of the name must be less than or equal to 31 characters.

Note: If you choose a *vsd_name* that is already the name of another device, the **cfgvsd** command will be unsuccessful for that virtual shared disk. This error ensures that the special device files created for the name do not overlay and destroy files of the same name representing some other device type (such as a logical volume).

nocache | cache
Affects how requests are processed at the server node. **nocache** is the default. **cache** tells the IBM Virtual Shared Disk software on the server node to use the cache for all 4KB requests on 4KB boundaries. Otherwise, the cache is not used.

The **cache** option should only be used if the using application gains performance by avoiding a 4KB read immediately after a 4KB write. Refer to *PSSP: Managing Shared Disks* for additional information on IBM Virtual Shared Disk tuning.

Description

This command is run to specify logical volumes residing on globally accessible volume groups to be used as virtual shared disks.

You can use the System Management Interface Tool (SMIT) to run the **defvsd** command. To use SMIT, enter:

```
smit vsd_data
```

and select the Define a Virtual Shared Disk option.

Security

You must be in the AIX **bin** group and have write access to the SDR to run this command.

Prerequisite Information

PSSP: Managing Shared Disks

Related Information

Commands: **vsdatalst**, **vsdvg**, **undefvsd**

Refer to *PSSP: Managing Shared Disks* for information regarding IBM Virtual Shared Disk performance enhancements.

Examples

1. The following example adds SDR information indicating that on globally accessible volume group **vg1n1**, the logical volume known as **lv1vg1n1** is used as a noncached virtual shared disk named **vsd1vg1n1**.

   ```
   defvsd lv1vg1n1 vg1n1 vsd1vg1n1
   ```

2. The following example defines cachable virtual shared disk **vsd1vg2n1** on the **lv2vg1n1** logical volume on the **vg1n1** globally accessible volume group

   ```
   defvsd lv2vg1n1 vg1n1 vsd1vg2n1 cache
   ```

delnimclient

Purpose

delnimclient – Deletes a Network Installation Management (NIM) client definition from a NIM master.

Syntax

delnimclient −**h** | −**l** *node_list* | −**s** *server_node_list*

Flags

−**h** Displays usage information. If the command is issued with the −**h** flag, the syntax description is displayed to standard output and no other action is taken (even if other valid flags are entered along with the −**h** flag).

−**l** *node_list* Indicates by *node_list* the SP nodes to be unconfigured as NIM clients of their boot/install servers. The *node_list* is a comma-separated list of node numbers.

−**s** *server_node_list*
Indicates by *server_node_list* the SP boot/install server nodes on which to delete all NIM clients that are no longer defined as boot/install clients in the System Data Repository (SDR). Server node 0 (zero) signifies the control workstation.

Operands

None.

Description

Use this command to undefine a node as a NIM client. This is accomplished by determining the node's boot/install server and unconfiguring that client node as a NIM client on that server. When complete, the entry for the specified client is deleted from the NIM configuration database on the server. This command does not change the boot/install attributes for the nodes in the System Data Repository.

Note: This command results in no processing on the client node.

Standard Error

This command writes error messages (as necessary) to standard error.

Exit Values

0 Indicates the successful completion of the command.

−**1** Indicates that an error occurred.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Location

/usr/lpp/ssp/bin/delnimclient

Related Information

Commands: **mknimclient**, **setup_server**

Examples

To delete the NIM client definition for nodes 1, 3, and 5 from the NIM database on their respective boot/install servers, enter:

```
delnimclient -l 1,3,5
```

delnimmast

Purpose

delnimmast – Unconfigures a node as a Network Installation Management (NIM) master.

Syntax

delnimmast **−h** | **−l** *node_list*

Flags

−h Displays usage information. If the command is issued with the **−h** flag, the syntax description is displayed to standard output and no other action is taken (even if other valid flags are entered along with the **−h** flag).

−l *node_list* Indicates by *node_list* the SP nodes to be unconfigured as NIM masters. The *node_list* is a comma-separated list of node numbers. Node number 0 (zero) signifies the control workstation.

Operands

None.

Description

Use this command to undefine a node as a NIM master. This command does not change the boot/install attributes for the nodes in the System Data Repository.

Standard Error

This command writes error messages (as necessary) to standard error.

Exit Values

0 Indicates the successful completion of the command.

−1 Indicates that an error occurred.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Location

/usr/lpp/ssp/bin/delnimmast

Related Information

Commands: **mknimmast**, **setup_server**

Examples

To unconfigure nodes 1, 3, and 5 as NIM masters and delete the NIM file sets, enter:

```
delnimmast -l 1,3,5
```

dsh

Purpose

dsh – Issues commands to a group of hosts in parallel.

Syntax

dsh [–**q**]

dsh [–**h**]

dsh [–**i**] [**v**] [**c**] [**a**] [**G**] [**d**] [**D**] [–**l** *login_name*] [–**N** *node_group*,*node_group*, ...] [–**w** {*host_names* | –}] [–**f** *fanout_value*] [*command*]

Flags

–**q** Displays a list of hosts in the current working collective file. The WCOLL environment variable is examined to find the name of the file containing the host names in a working collective, and host names from that file are displayed. In addition, the value of the FANOUT environment variable is displayed.

–**h** Displays usage information.

–**i** Contains information about the working collective and commands. If this flag is set, the working collective and the command is displayed as each command is issued.

–**v** Verifies hosts before adding to the working collective. If this flag is set, each host to be added to the working collective is checked before it is added to the collective. If a host is not responding, it is not included in the working collective. In command line mode you will be prompted to include any host which is not responding. Anything other than "Y" or "y" will result in the host being excluded from the working collective.

–**c** Indicates that **dsh** continues to send commands to hosts for which previous **rsh** runs have returned a nonzero return code. If this flag is not set, the host is removed from the working collective for the duration of this **dsh** command.

–**a** Specifies that the System Data Repository **initial_hostname** field for all nodes in the current system partition be added to the working collective. If –**G** is specified, all nodes in the SP system are included.

–**G** Changes the scope of the –**a** and –**N** arguments from the current system partition to the SP system.

–**d** Forwards the DCE credentials for authentication by using the **-f** flag on the **rsh** command. If this flag is not set, the **dsh** command will not forward DCE credentials.

–**D** Recursively forwards the DCE credentials for authentication by using the **-F** flag on the **rsh** command. If this flag is not set, the **dsh** command will not forward DCE credentials.

–**l** Specifies a remote user name under which to execute the commands. If **l** is not used, the remote user name is the same as your local user name. (This is lowercase **l**, as in **l**ist.)

−w Specifies a list of host names, separated by commas, to include in the working collective. Both this flag and the **a** flag can be included on the same command line. If "−" is specified, host names are read from standard input. If **−w** − is used, commands cannot be read from standard input. Duplicate host names are only included once in the working collective.

−f Specifies a fanout value. The default value is 64. It indicates the maximum number of concurrent **rsh** runs to execute. Sequential execution can be specified by indicating a fanout value of 1. The fanout value is taken from the FANOUT environment variable if the **f** flag is not specified, otherwise the default is used.

−N Specifies a list of node groups. Each node group is resolved into nodes and these nodes are added to the working collective. If **−G** is supplied, a global node group is used. Otherwise, a partitioned-bound node group is used.

If the −**a**, −**w**, or −**N** flags are not specified, the WCOLL environment variable contains the name of a file containing host names for the working collective.

Operands

command Specifies a command to execute on the working collective. It is passed to **rsh**. This command is specified in **rsh** syntax.

Description

Note: The **dsh** command relies on the AIX **rsh** command to function. With AIX 4.3.1 **rsh** has the capability of supporting DCE in addition to Kerberos Version 4 on the SP. DCE credentials may be required depending on the authentication method set on the host. The AIX command **lsauthent** or the SP command **lsauthpar** can be used to find the authentication method currently in use.

The **dsh** executes commands against all or any subset of the hosts in a network. If the command operand is not specified, it reads lines from the command line or standard input and executes each as a command on a set of network-connected hosts. To exit the **dsh** command line mode type "exit" or press enter at the **dsh>** prompt. Alternatively, a single command in **rsh** syntax can be specified on the **dsh** command line.

As each command is read, it is interpreted by passing it to each host in a group called the *working collective*.

The working collective is obtained from the first existence of one of the following:

1. A list of host names specified on the command line and the members of the cluster as listed in the System Data Repository.
2. The contents of a file named by the WCOLL environment variable.

If neither of these exist, an error has occurred and no commands are issued.

The working collective file should have one host name per line. Blank lines and comment lines beginning with **#** are ignored.

The path used when resolving the **dsh** command on the target nodes is the path set by the user with the DSHPATH environment variable. If DSHPATH is not set, the path used is the **rsh** default path, **/usr/ucb:/bin:/usr/bin:**. The DSHPATH environment variable only works when the user's remote login shell is the Bourne or Korn shell. An example would be to set DSHPATH to the path set on the source machine (for example, DSHPATH=$PATH).

The maximum number of concurrent **rsh** runs can be specified with the fanout (**f**) flag or via the FANOUT environment variable. If desired, sequential execution can be obtained by specifying a fanout value of 1. Results are displayed as remote commands complete. All **rsh** runs in a fanout must complete before the next set of **rsh** runs is started. If fanout is not specified via FANOUT or the **f** flag, **rsh** runs to 64 hosts are issued concurrently. Each **rsh** that **dsh** runs requires a reserved TCP/IP port and only 512 such ports are available per host. With large fanouts, it is possible to exhaust all the ports on a host, causing commands that use these ports to be unsuccessful.

Exit values for the **rsh** commands are displayed in messages from **dsh** if nonzero. (A nonzero return code from **rsh** indicates that the **rsh** was unsuccessful; it has nothing to do with the exit code of the remotely executed command.) If an **rsh** is unsuccessful, that host is removed from the current working collective (not the current working collective file), unless the **c** flag was set.

The **dsh** exit value is 0 if no errors occurred in the **dsh** command and all **rsh** runs finished with exit codes of 0. The **dsh** exit value is more than 0 if internal errors occur or the **rsh** runs are unsuccessful. The exit value is increased by 1 for each **rsh** error.

No particular error recovery for command errors on remote hosts is provided. The application or user can examine the command results in **dsh**'s standard error and standard output, and take appropriate action.

The **dsh** command waits until results are in for each command for all hosts and displays those results before reading more input commands.

The **dsh** command does not work with interactive commands, including those that read from standard input.

The **dsh** command output consists of the output (standard error and standard output) of the remotely executed commands. The **dsh** standard output is the standard output of the remote command. The **dsh** standard error is the standard error of the remote command. Each line is prefixed with the host name of the host from which that output came. The host name is followed by ":" and a line of the command output.

For example, let's say that a command was issued to a working collective of host1, host2, and host3. When the command was issued on each of the hosts, the following lines were written by the remote commands:

```
For host1 stdout:

h1out1

h1out2


For host2 stdout:

h2out1

h2out2


For host3 stdout:

h3out1


For host3 stderr:

h3err1

h3err2


dsh stdout will be

host1: h1out1

host1: h1out2

host2: h2out1

host2: h2out2

host3: h3out1


dsh stderr will be

host3: h3err1

host3: h3err2
```

A filter to display the output lines by the host is provided separately. See the **dshbak** command.

If a host is detected as down (for example, an **rsh** returns a nonzero return code), subsequent commands are not sent to it on this invocation of **dsh**, unless the **c** (continue) option is specified on the command line.

An exclamation point at the beginning of a command line causes the command to be passed directly to the local host in the current environment. The command is not sent to the working collective.

Signals 2 (INT), 3 (QUIT), and 15 (TERM) are propagated to the remote commands.

Signals 19 (CONT), 17 (STOP), and 18 (TSTP) are defaulted. This means that the **dsh** command responds normally to these signals, but they do not have an effect on the remotely running commands. Other signals are caught by **dsh** and have their default effects on the **dsh** command. In the case of these other signals, all current children, and via propagation their remotely running commands, are terminated (SIGTERM).

Files

/usr/sbin/dshbak
The supplied backend formatting filter.

working collective file
A file containing host names, one per line, that defines a working collective.

Security

You must have access to the AIX Secure Remote Commands to run this command.

Restrictions

The command attempts to execute the command string on the remote node, using the current operating locale of the process on the local node from which the command was issued. If this locale is not installed on the remote node, default AIX behavior is used, such that the remote command executes in the default locale defined to AIX on that node.

Location

/usr/sbin/dsh

Related Information

PSSP Commands: **dshbak**, **sysctl**

AIX Commands: **rsh**

Examples

1. To issue the **ps** command on each host listed in the **wchosts** file, enter:

```
WCOLL=./wchosts dsh ps
```

2. To list the current working collective file as specified by the WCOLL environment variable, enter:

```
dsh -q
```

3. To set the working collective to three hosts and start reading commands from standard input, enter:

```
dsh -w otherhost1,otherhost2,otherhost3
```

4. To set the current working collective to three hosts, plus the members of the cluster, and issue a command on those hosts formatting the output, enter:

   ```
   dsh -w host1,host2,host3 -a cat /etc/passwd | dshbak
   ```

5. To append the file remotefile on otherhost to otherremotefile, which is on otherhost, enter:

   ```
   dsh -w otherhost cat remotefile '>>' otherremotefile
   ```

6. To run a file of commands sequentially on all the members of the current system partition and save the results in a file, including the collective and the working collective for each command, enter:

   ```
   dsh -if 1 -a < commands_file > results 2>&1
   ```

7. To run the **ps** command on the working collective and filter results locally, enter:

   ```
   dsh ps -ef | grep root
   ```

8. To run the **ps** command and filter results on the working collective hosts (this can improve performance considerably), enter:

   ```
   dsh 'ps -ef | grep root'
   ```

   or

   ```
   dsh ps -ef "|" grep root
   ```

9. To cat a file from host1 to the local system stripping off the preceding host name to preserve the file, enter:

   ```
   dsh -w host1 cat /etc/passwd | cut -d: -f2- | cut -c2- > myetcpasswd
   ```

10. To run the **ps** command on each node in the node group my_nodes, enter:

    ```
    dsh -N my_nodes ps
    ```

11. To run the **rm** command to remove **/tmp/error.log** on otherhost using the DCE credentials of this user, enter:

    ```
    dsh -D otherhost /bin/rm /tmp/error.log
    ```

dshbak

Purpose

dshbak – Presents formatted output from the **dsh** and **sysctl** commands.

Syntax

dshbak [–**c**]

Flags

–**c** Collapses identical output from more than one host so that it is displayed only once.

Operands

None.

Description

The **dshbak** command takes lines in the following format:

```
host_name: line of output from remote command
```

The **dshbak** command formats them as follows and writes them to standard output. Assume that the output from host_name3 and host_name4 is identical and the **c** option was specified:

```
HOSTS -----------------------------------------------------------------

host_name1

-----------------------------------------------------------------------

.

.

lines from dsh or sysctl with host_names stripped off

.

.

HOSTS -----------------------------------------------------------------

host_name2

-----------------------------------------------------------------------

.

.

lines from dsh or sysctl with host_names stripped off

.

.

HOSTS -----------------------------------------------------------------

host_name3             host_name4

-----------------------------------------------------------------------

.

.

lines from dsh or sysctl with host_names stripped off

.

.
```

When output is displayed from more than one host in collapsed form, the host names are displayed alphabetically.

When output is not collapsed, output is displayed sorted alphabetically by host name.

The **dshbak** command writes "." for each 1000 lines of output filtered.

Standard Error

When the **dshbak** filter is used, and stderr msgs are generated, then all error msgs (on stderr) will appear before all stdout msgs. This is true with and without **-c** specified. For example,

```
[k7s][/]> dsh -w k7n01,k7n08 lsauthent | dshbak -c

k7n08: rshd: Kerberos 5 Authentication Failed: User hosts/k7cw.ppd.pok.

ibm.com/self@k7dcecell is not authorized to login to account root.

k7n08: rshd: Kerberos Authentication Failed: Access denied because of

improper credentials.

k7n08: spk4rsh: 0041-004 Kerberos rcmd failed: rcmd protocol failure.

HOSTS -------------------------------------------------------------------

k7n01              k7n08

-------------------------------------------------------------------------

Kerberos 5

Kerberos 4

Standard Aix
```

Location

/usr/sbin/dshbak

Related Information

Commands: **dsh**, **sysctl**

Examples

1. To display the results of a command executed on several hosts in the format described previously, enter:

   ```
   dsh -w host1,host2,host3 cat /etc/passwd | dshbak
   ```

2. To display the results of a command executed on several hosts with identical output displayed only once, enter:

   ```
   dsh -w host1,host2,host3 pwd | dshbak -c
   ```

dsrvtgt

Purpose

dsrvtgt – Gets DCE credentials as a service principal, or as a user using a keyfile.

Syntax

dsrvtgt {–**h** | [–**f**] *service-name* | *user–principal keytab–path*}

Flags

–h Specifies that the command syntax is to be listed. When this flag is specified, any operands are ignored.

–f Specifies that the credentials that can be forwarded should be obtained, if possible.

Operands

service-name
The predefined name of a service listed in the **spsec_defaults** file, if the client will be a service principal.

user–principal
The DCE principal name of the user, whose identity will be used to get credentials.

keytab–path
The full pathname of the keytab file to use when specifying a user principal name. This operand is not used when a service–name is specified.

Description

The **dsrvtgt** command uses the private key of a principal stored in a keytab file to obtain credentials as a DCE principal, rather than a password. It prints to STDOUT the value that must be assigned to the KRB5CCNAME environment variable to allow the credentials to be used by the invoking process. It is the responsibility of the calling process to remove the credentials using the **kdestroy** command.

If **Kerberos 5** is not an active authentication method for AIX remote commands and DCE is not an active authentication method for SP trusted services, this command performs no function and generates no output, but returns successfully.

Files

The credentials cache file created by DCE in **/opt/dcelocal/var/security/creds**.

The **spsec_defaults** configuration file located in **/usr/lpp/ssp/config**.

The **spsec_overrides** configuration file located in **/spdata/sys1/spsec**.

A keytab file. For a service principal, it will be named **/spdata/sys1/keyfiles/principal**.

Standard Output

If DCE credentials are obtained, this command writes the name of the credentials cache file that was created prefixed by **"FILE:"**.

Standard Error

Output consists of error messages, when the command cannot complete successfully.

There are no unique consequences of command errors.

Exit Values

0 Indicates successful completion of the command.

1 Indicates that an error occurred.

Security

The **dsrvtgt** command can be used by any user who owns a keytab file. When the operand is a *service–name*, the command must be run by the AIX userid under which the server daemon runs. For example, to log in as a service principal whose keyfile is owned by *root*, you must be *root*.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP) (fileset **ssp.clients**).

Prerequisite Information

The online AIX DCE publications for administrators and users.

The chapters on security in the *PSSP: Administration Guide*.

Location

/usr/lpp/ssp/bin/dsrvtgt

Related Information

Commands: **spmankey**

Examples

1. This **ksh** fragment shows the use of **dsrvtgt** in a script to obtain credentials using the identity of the generic background root service:

```
# Get DCE credentials and set KRB5CCNAME, if required
nkrb5=$(/bin/dsrvtgt ssp/spbgroot 2>&1)
if [[ $? —ne 0 ]] then
print ''$nkrb5''
nkrb5=''''
fi
if [[  ''$nkrb5'' != '''' ]] then
okrb5=$KRB5CCNAME
export KRB5CCNAME=$nkrb5
fi

# Use the credentials to access the SDR
SDRChangeAttrValues ..

# Get rid of DCE credentials after use
if [[ ''$nkrb5'' !=      ]] then
/bin/kdestroy >/dev/null 2>&1
if [[ ''$okrb5'' !=      ]] then
export KRB5CCNAME=$okrb5
else
unset KRB5CCNAME
fi
fi
```

2. Obtaining credentials as an ordinary user principal to invoke a trusted service client interface from a background script; in this case the DCE principal ralph must have previously created the keytab file using his current DCE password:

```
export KRB5CCNAME=$(/bin/dsrvtgt ralph /home/ralph/keyfile)
```

Eannotator

Purpose

Eannotator – Annotates the connection labels in the topology file.

Syntax

Eannotator [–**h**] {–**F** *input_file* | –**d**} –**f** *output_file* –**O** [**yes | no**] [–**p** {**0 | all**}]

Flags

–**h** Displays usage information.

–**d** Specifies that the topology filename is to be determined from the contents of the SDR, based on the number and type of switches in the system. On an SP Switch system, this flag is only valid on a system which has not been partitioned.

–**F** Specifies the topology input file.

–**f** Specifies the topology output file.

–**O** Specifies whether to save the output file to the System Data Repository (SDR) or to the current directory. **yes** saves the output file to the SDR via the **Etopology** command. **no** saves the output file to the current directory.

–**p {0 | all}**
Specifies for which switch plane the operation is to be performed. If not specified, the default is to perform the operation for all valid switch planes. The only valid switch plane value is 0. This flag is valid only on systems with SP Switch2 switches.

Operands

None.

Description

This command supports all of the following:

- The SP Switch
- The SP Switch-8
- The SP Switch2

This command must be executed whenever a new topology file is selected.

The topology file contains node-to-switch or switch-to-switch cable information. A node-to-switch connection looks like following:

```
s 25 2 tb0 17 0     E2-S00-BH-J16 to E2-N2
```

The predefined node-to-switch connections start with an “s” which indicates a switch connection. The next two digits, in this case “25” indicate the switch (2) and switch chip (5) being connected. The next digit, in this case “2”, indicates the switch chip port in the connection. The next field, in this case “tb0”, specifies the type of adapter present in the SP node. The following field, in this case “17”, is the switch node number for the SP node, and the last digit, in this case “0”, indicates the adapter port within the connection.

For switch-to-switch connections, the first four fields (switch indicator, switch, switch chip, and switch chip port) are repeated to identify the other end of the connection.

The connection label “E2-S00-BH-J16 to E2-N2” provides physical connection information for a customer's use to identify the problem connection.

Depending on the customer's physical switch frame configuration defined in the SDR, the **Eannotator** command retrieves switch node and dependent node objects from the SDR and applies proper connection information to the topology file.

If the input topology file contains existing connection information, the **Eannotator** command replaces the existing connection label with the new connection labels. If the input topology file does not contain connection labels, the **Eannotator** command appends the proper connection label to each line on the topology file.

The precoded connection labels on the topology file start with an “L” which indicate logical frames. The **Eannotator** command replaces the “L” character with an “E” which indicates physical frames. The “S” character indicates which slot the switch occupies in the frame, the “BH” characters indicate a Bulk Head connection, the “J” character indicates which jack provides the connection from the switch board, the “N” character indicates the node being connected to the switch, and the “SC” characters indicate the Switch Chip connection.

If you have a partitioned system and need to do a reannotate, you will need to make sure that you are reannotating the correct topology file.

1. Make sure you are in the correct partition first by exporting the correct partition name. For example:

   ```
   export SP_NAME=partition_name
   ```

2. Run Etopology to get the correct topology file. For example:

   ```
   Etopology -read /tmp/temporary.top
   ```

3. Run Eannotator with option **-O yes** to save output file to SDR via Etopology command. For example:

   ```
   Eannotator -F /tmp/temporary.top -f /tmp/expected.top.annotate -O yes
   ```

Files

/etc/SP/expected.top.1nsb_8.0isb.0
: The standard topology file for systems with a maximum of eight nodes.

/etc/SP/expected.top.1nsb.0isb.0
: The standard topology file for one NSB system or a maximum of 16 nodes.

/etc/SP/expected.top.2nsb.0isb.0
: The standard topology file for two NSB systems or a maximum of 32 nodes.

/etc/SP/expected.top.3nsb.0isb.0
The standard topology file for three NSB systems or a maximum of 48 nodes.

/etc/SP/expected.top.4nsb.0isb.0
The standard topology file for four NSB systems or a maximum of 64 nodes.

/etc/SP/expected.top.5nsb.0isb.0
The standard topology file for five NSB systems or a maximum of 80 nodes.

/etc/SP/expected.top.5nsb.4isb.0
The standard topology file for five NSB and four ISB systems or a maximum of 80 nodes. This is an advantage-type network with a higher bisectional bandwidth.

/etc/SP/expected.top.6nsb.4isb.0
The standard topology file for six NSB and four ISB systems or a maximum of 96 nodes.

/etc/SP/expected.top.7nsb.4isb.0
The standard topology file for seven NSB and four ISB systems or a maximum of 112 nodes.

/etc/SP/expected.top.8nsb.4isb.0
The standard topology file for eight NSB and four ISB systems or a maximum of 128 nodes.

/etc/SP/expected.top.1nsb_8.0isb.1
The standard topology file for systems with an SP Switch-8 and a maximum of eight nodes.

Security

You must have root privilege to run this command.

Location

/usr/lpp/ssp/bin/Eannotator

Related Information

Commands: **Eclock**, **Efence**, **Eprimary**, **Equiesce**, **Estart**, **Etopology**, **Eunfence**, **Eupartition**

Refer to *IBM RS/6000 SP: Planning, Volume 2, Control Workstation and Software Environment* for details about system partition topology files.

Examples

1. The following are the topology file entries before and after the **Eannotator** command executes:

```
For SP Switch

Before:

s 15 3 tb3 0 0 L01-S00-BH-J18 to L01-N1


After:

s 15 3 tb3 0 0 E01-S17-BH-J18 to E01-N1


For SP Switch2

Before:

s 15 3 tb3 0 0 L01-S00-BH-J18 to L01-N1


After:

s 15 3 xxx x x E01-S17-BH-J18 to E01-N1
```

Note: Logical frame L01 is defined as physical frame 1 in the SDR Switch object.

```
Before:

s 10016 0 s 51 3 L09-S1-BH-J20 to L05-S00-BH-J19


After:

s 10016 0 s 51 3 E10-S1-BH-J20 to E05-S17-BH-J19
```

Note: Logical frame L09 is defined as physical frame 10 in the SDR Switch object.

```
Before:

s 15 3 tb0 0 0 L03-S00-BH-J18 to L03-N3


After:

s 15 3 tb3 0 0 E03-S17-BH-J18 to E03-N3 # Dependent Node
```

Note: Logical frame L03 is defined as physical frame 3 in the SDR Switch object and the node was determined to be a dependent node.

2. To annotate a topology file for a 128-way SP system with eight Node Switch Boards (NSBs) and four Intermediate Switch Boards (ISBs) and to save the output file in the current directory, enter:

```
Eannotator -F expected.top.8nsb.4isb.0 -f expected.top -O no
```

3. To annotate an SP Switch2 system with a system determined topology file, enter:

   ```
   Eannotator -d -f expected.top -O yes
   ```

4. To annotate a topology file for a 16-way SP system with one NSB and no ISBs and to save the output file in the SDR via the **Etopology** command, enter:

   ```
   Eannotator -F expected.top.1nsb.0isb.0 -f expected.top -O yes
   ```

5. To reannotate a partitioned system, you will need to get the correct topology file from the **Etopology** command.

   a. Make sure you are in the correct partition first by exporting the correct partition name:

      ```
      export SP_NAME=partition_name
      ```

   b. Run **Etopology** to get the correct topology file:

      ```
      Etopology -read /tmp/temporary.top
      ```

   c. Run **Eannotator** with option **-O yes** to save output file to SDR via **Etoplogy** command:

      ```
      Eannotator -F /tmp/temporary.top -f /tmp/expected.top.annotate -O yes
      ```

Eclock

Purpose

Eclock – Controls the clock source for each switch board within an SP cluster.

Syntax

Eclock [−**f** *Eclock_topology_file*] | [−**a** *Eclock_topology_file*] | [−**r**] | [−**d**] |
[−**s** *switch_number* −**m** *mux_value*] | [−**c** *Eclock_topology_file*]

Flags

−**f** *Eclock_topology_file*
: Specifies the file name of the clock topology file containing the initial switch clock input values for all switches in the system.

−**a** *Eclock_topology_file*
: Uses the alternate **Eclock** topology specified in the given clock topology file.

−**r**
: Extracts the clock topology file information from the System Data Repository (SDR) and initializes the switch clock inputs for all switches in the system.

−**d**
: Detects the switch configuration, automatically selects the clock topology file, and initializes the switch clock inputs for all switches in the system.

−**s** *switch_number* −**m** *mux_value*
: Sets an individual switch (*switch_number*) clock multiplexor (mux) value (*mux_value*)

where:

switch_number Specifies the switch number.

mux_value Specifies a flag with one of the following values:

- **0** Use the internal oscillator (make this frame the master frame).
- **1** Use input 1 (clock input from jack 3) (NSBs or ISBs).
- **2** Use input 2 (clock input from jack 4) (NSBs or ISBs).
- **3** Use clock input from jack 5 (NSBs or ISBs).
- **4** Use clock input from jack 4 (NSBs or ISBs).
- **5** Use clock input from jack 5 (NSBs or ISBs).
- **6** Use clock input from jack 6 (NSBs or ISBs).
- **7** Use clock input from jack 7 (ISBs only).
- **8** Use clock input from jack 8 (ISBs only).
- **9** Use clock input from jack 9 (ISBs only).
- **10** Use clock input from jack 10 (ISBs only).
- **27** Use clock input from jack 27 (NSBs or ISBs).
- **28** Use clock input from jack 28 (NSBs or ISBs).
- **29** Use clock input from jack 29 (NSBs or ISBs).
- **30** Use clock input from jack 30 (NSBs or ISBs).
- **31** Use clock input from jack 31 (ISBs only).

32 Use clock input from jack 32 (ISBs only).
33 Use clock input from jack 33 (ISBs only).
34 Use clock input from jack 34 (ISBs only).

–**c** *Eclock_topology_file*
Creates a new clock topology file from the data in the SDR.

If a flag is not specified, the clock input values stored in the SDR are displayed.

Operands

None.

Description

Note: This command is not valid on a system with an SP Switch2 switch.

Use this command to set the multiplexors that control the clocking at each switch board within the configuration. One switch board within the configuration is designated as the "Master" switch that provides the clocking signal for all other switch boards within the configuration. The **Eclock** command reads clock topology information from either the file specified on the command line or the clock topology data within the SDR. If a clock topology file was specified, the **Eclock** command places the clock topology information into the SDR, so that it can be accessed again during a subsequent **Eclock** invocation. After processing the clock topology file, **Eclock** causes the new clock topology to take effect for the switches specified. A clock topology file contains the following information for each switch board within the cluster:

- A switch board number.
- A flag indicating the clock source for the switch within the frame. The flag can have any of the following settings:

  0 Use the internal oscillator (make this frame the master frame).
  1 Use input 1 (clock input from jack 3) (NSBs or ISBs).
  2 Use input 2 (clock input from jack 4) (NSBs or ISBs).
  3 Use clock input from jack 5 (NSBs or ISBs).
  4 Use clock input from jack 4 (NSBs or ISBs).
  5 Use clock input from jack 5 (NSBs or ISBs).
  6 Use clock input from jack 6 (NSBs or ISBs).
  7 Use clock input from jack 7 (ISBs only).
  8 Use clock input from jack 8 (ISBs only).
  9 Use clock input from jack 9 (ISBs only).
  10 Use clock input from jack 10 (ISBs only).
  27 Use clock input from jack 27 (NSBs or ISBs).
  28 Use clock input from jack 28 (NSBs or ISBs).
  29 Use clock input from jack 29 (NSBs or ISBs).
  30 Use clock input from jack 30 (NSBs or ISBs).
  31 Use clock input from jack 31 (ISBs only).
  32 Use clock input from jack 32 (ISBs only).
  33 Use clock input from jack 33 (ISBs only).
  34 Use clock input from jack 34 (ISBs only).

- The receiving jack number (*xx* if this switch board is the master).
- The switch board number providing the clocking signal (zero if this switch board is the master).

- The clock source jack number.
- An alternate clock source (mux) value.
- The alternate clock receiver jack number.
- The switch board number providing the alternate clock source (mux) value.
- The alternate sending jack number (*xx* if this switch board is the master).

SP Switch Warning

If **Eclock** is run to change the clock multiplexor settings while the switch is operational, you will experience **css** outages until a subsequent **Estart** is completed. If you run **Eclock** and specify the **-f**, **-a**, **-r** or **-d** flag, you do not need to run **Estart** if the swtadmd subsystem is active. In this case the subsystem runs **Estart** for you.

SP Switch Considerations

Eclock on the SP switch recycles the **fault_service_Worm_RTG_SP** (Worm) daemon on the nodes. If the switch was operational when the **Eclock** command was issued, you must run the **Estart** command following the switch adjustment.

Since **Eclock** operates across system partitions, if you specified the **-f**, **-a**, **-r** or **-d** flag, you must run the **Estart** command in ALL system partitions unless the swtadmd subsystem is active. In this case the subsystem runs **Estart** for you. Since the **-s** flag operates just on the specified switch, you need to run **Estart** only in the partitions which share that switch. However, if you used the **-s** command to reset the master switch, the effect is the same as having issued a global **Eclock** command and you must run **Estart** in all partitions. The **-s** option will recycle the Worm daemons only on the nodes connected to the target switches.

The **-s** option will recycle the Worm daemons only on the nodes connected to the target switches. There are certain considerations which must be taken into account when running **Eclock -s**. Assuming the master switch is not one of the target switches:

- If the primary node is not on a targeted switch, its Worm daemon will not be recycled. However, the primary's link to other nodes over the switch will be disrupted. Attempts to **Eunfence** nodes may timeout or cause the Worm daemon to die on other nodes. To avoid this, run **Equiesce** prior to running the **Eclock -s** command. After the **Eclock**, run the **Estart** command to get a functional primary node. If necessary, run the **Eunfence** command to bring isolated nodes on the switch.
- If the primary node is on a targeted switch but the primary backup is not, you can wait for primary node takeover to complete. A new primary backup node will be chosen. If necessary, run the **Eunfence** command to bring isolated nodes on the switch.
- If both the primary and primary backup nodes are on a targeted switch, the Worm daemon on both these nodes will be recycled, and you must run **Estart** to get a functional primary node. If necessary, run the **Eunfence** command to bring isolated nodes on the switch.

Files

/etc/SP/Eclock.top.1nsb.0isb.0
The standard clock topology file for systems with one NSB or a maximum of 16 nodes.

/etc/SP/Eclock.top.1nsb_8.0isb.0
The standard clock topology file for systems with an SP Switch-8 or a maximum of eight nodes.

/etc/SP/Eclock.top.2nsb.0isb.0
The standard clock topology file for systems with two NSBs or a maximum of 32 nodes.

/etc/SP/Eclock.top.3nsb.0isb.0
The standard clock topology file for systems with three NSBs or a maximum of 48 nodes.

/etc/SP/Eclock.top.4nsb.0isb.0
The standard clock topology file for systems with four NSBs or a maximum of 64 nodes.

/etc/SP/Eclock.top.5nsb.0isb.0
The standard clock topology file for systems with five NSBs or a maximum of 80 nodes.

/etc/SP/Eclock.top.5nsb.4isb.0
The standard clock topology file for systems with five NSBs and four ISBs or a maximum of 80 nodes. This is an advantage-type network with a higher bisectional bandwidth.

/etc/SP/Eclock.top.6nsb.4isb.0
The standard clock topology file for systems with six NSBs and four ISBs or a maximum of 96 nodes.

/etc/SP/Eclock.top.7nsb.4isb.0
The standard clock topology file for systems with seven NSBs and four ISBs or a maximum of 112 nodes.

/etc/SP/Eclock.top.8nsb.4isb.0
The standard clock topology file for systems with eight NSBs and four ISBs or a maximum of 128 nodes.

Security

You must have root privilege to run this command.

Restrictions

This command is not valid on a system with an SP Switch2 switch.

Location

/usr/lpp/ssp/bin/Eclock

Related Information

Commands: **Eannotator**, **Efence**, **Eprimary**, **Equiesce**, **Estart**, **Etopology**, **Eunfence**, **Eunpartition**

Examples

1. To set the clock multiplexors for a 128-way SP system with eight Node Switch Boards (NSBs) and four Intermediate Switch Boards (ISBs), enter:

   ```
   Eclock -f /etc/SP/Eclock.top.8nsb.4isb.0
   ```

2. To display the clock multiplexor settings for all switches within the SP system, enter:

   ```
   Eclock
   ```

3. To set the switch on frame 1 (switch 1) to be the master switch (use internal oscillator), enter:

   ```
   Eclock -s 1 -m 0
   ```

4. To create an **Eclock** topology file from the current data in the SDR, enter:

```
Eclock -c /tmp/Eclock.top
```

5. To use an alternate clock topology (with a new switch clock source) for a 64-way SP system with two ISBs, enter:

```
Eclock -a /etc/SP/Eclock.top.4nsb.2isb.0
```

6. To have **Eclock** automatically select a topology file for you based on data in the SDR, enter:

```
Eclock -d
```

Efence

Purpose

Efence – Removes an SP node from the current *active* switch network.

Syntax

Efence [−**h**] | [−**G**] [−**autojoin**] [−**f**] [−**p** {**0** | **all**}] [*node_specifier*] ...

Flags

−h Displays usage information.

−G Fences all valid nodes in the list of nodes regardless of system partition boundaries. If the −**G** flag is not used, the **Efence** command will only fence the nodes in the current system partition. All other specified nodes will not be fenced and a nonzero return code is returned.

−autojoin

With PSSP 3.1 or later releases, you can choose to have nodes automatically unfenced. With the automatic unfence feature enabled, whenever a node reboots it will automatically join the switch. Also, a node which is Efenced with autojoin will automatically join the switch within two minutes due to the new automatic unfence function in PSSP 3.1 or later releases.

If the automatic unfence feature is disabled, or in a coexistence environment with the primary node at PSSP 2.4 or earlier, the **autojoin** option enables the nodes in the argument list to be fenced and to automatically rejoin the current switch network if the node is rebooted or the Fault Service daemon is restarted.

If you have an SP Switch installed on your system, such nodes are also rejoined when an **Estart** command is issued.

The default for PSSP 3.1 or later releases, is to have the automatic unfence feature enabled. See the **Estart** command for how to change the default.

−p {0 | all}
Specifies for which switch plane the operation is to be performed. If not specified, the default is to perform the operation for all valid switch planes. The only valid switch plane value is 0. This flag is valid only on systems with SP Switch2 switches.

−f Allows the current primary and/or primary backup to be fenced. If not specified, the current primary and primary backup may not be fenced. If a current primary node is specified to be fenced, the **Efence** command will run **Eprimary** to make the current primary backup node the new oncoming primary node, turn on the isolated bit for the old primary node, and then run **Estart**. If a current backup node is specified to be fenced, **Efence** will turn on the isolated bit for the old primary backup node and run **Estart**. If both the current primary and the current primary backup node are specified to be fenced, a new oncoming primary node is selected, **Eprimary** is run, the isolated bit for the old primary and primary backup node is turned on, and **Estart** is run.

Operands

node_specifier

Specifies a node or a list of nodes that are to be taken out of the current switch network. It can be a list of host names, IP addresses, node numbers, frame,slot pairs, or a node group.

Note: You cannot fence either the primary or primary backup nodes on the SP Switch, unless you specify the –**f** flag.

Description

Use the **Efence** command to remove a node from being part of the current switch network. Once a node is fenced, it cannot communicate with other nodes on the switch network, nor cause errors on the network. This command should be used (without the **-autojoin** flag) if you want to isolate the node for a period of time, for example, for service or maintenance. To bring the node back on the switch network, use the **Eunfence** command.

Note: If a host name or IP address is used as the *node_specifier* for a dependent node, it must be a host name or IP address assigned to the adapter that connects the dependent node to the SP Switch. Neither the administrative host name nor the Simple Network Management Protocol (SNMP) agent's host name for a dependent node is guaranteed to be the same as the host name of its switch network interface.

Security

You must have root privilege to run this command.

Location

/usr/lpp/ssp/bin

Related Information

Commands: **Eannotator**, **Eclock**, **Eprimary**, **Equiesce**, **Estart**, **Etopology**, **Eunfence**, **Eunpartition**

Examples

1. To display all the nodes that were fenced from the switch network in the current system partition, enter:

   ```
   Efence
   ```

2. To display all the nodes that were fenced from the switch network in all system partitions, enter:

   ```
   Efence -G
   ```

3. To fence two nodes by IP address, enter:

   ```
   Efence 129.33.34.1 129.33.34.6
   ```

4. To fence a node by host name, enter:

   ```
   Efence r11n01
   ```

5. To fence node 14 of frame 2 by frame,slot pair, enter:

```
Efence 2,14
```

6. If the current partition has nodes with node numbers 1, 2, 5, and 6 and another partition has nodes with node numbers 3, 4, 7, and 8, issuing the command:

   ```
   Efence 5 6 7 8
   ```

   fences nodes 5 and 6, but not nodes 7 and 8. As a result, the command returns a nonzero return code.

7. To successfully fence the nodes in example 6 with the same partitions, use the –**G** flag as follows:

   ```
   Efence -G 5 6 7 8
   ```

Emaster

Purpose

Emaster – Displays the master switch sequencing (MSS) node.

Syntax

Emaster [−**h**]

Flags

−**h** Displays usage information.

Operands

None.

Description

The MSS node is the node which periodically re−synchronizes time−of−day (TOD) signals on the SP Switch2. Use the **Emaster** command to display the node number of the MSS node.

Standard Output

Output consists of informational messages indicating the current MSS node.

Standard Error

Output consists of error messages, when the command cannot complete successfully.

Exit Values

0 Indicates successful completion of the command.

1 Indicates that an error occurred.

Security

You must have root privilege to run this command.

Restrictions

The **Emaster** command may be used only on systems with SP Switch2 switches.

Location

/usr/lpp/ssp/bin/Emaster

Related Information

The **emasterd** daemon (SRC subsystem name **emaster**) monitors the health and connectivity of the MSS node. If it goes down or loses connectivity to the switch, the **emasterd** daemon changes the MSS node.

Examples

To display the node number of the current MSS node, enter:

```
Emaster
```

emconditionctrl Script

Purpose

emconditionctrl – Loads the System Data Repository (SDR) with predefined Event Management conditions.

Syntax

emconditionctrl [–**a**] [–**s**] [–**k**] [–**d**] [–**c**] [–**t**] [–**o**] [–**r**] [–**h**]

Flags

–**a** Loads the SDR with predefined Event Management conditions for the current system partition.

–**s** Starts the subsystem. (Currently has no effect.)

–**k** Stops the subsystem. (Currently has no effect.)

–**d** Deletes the subsystem. (Currently has no effect.)

–**c** Cleans the subsystem. (Currently has no effect.)

–**t** Turns tracing on. (Currently has no effect.)

–**o** Turns tracing off. (Currently has no effect.)

–**r** Refreshes the subsystem. (Currently has no effect.)

–**h** Displays usage information.

Operands

None.

Description

The **emconditionctrl** script loads the SDR with some useful conditions that can be used for registering for Event Management events. Currently the SP Perspectives application can make use of conditions.

The **emconditionctrl** script is not normally executed on the command line. It is normally called by the **syspar_ctrl** command after the control workstation has been installed or when the system is partitioned. It implements all of the flags that **syspar_ctrl** can pass to its subsystems, although only the –**a** flag causes any change to the system. The –**a** flag causes predefined conditions to be loaded only if run on the control workstation. It has no effect if run elsewhere.

Exit Values

0 Indicates the successful completion of the command.

nonzero Indicates an exit code from the **SDRCreateObjects** command.

Security

You must have root privilege and write access to the SDR to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Location

/usr/lpp/ssp/bin/emconditionctrl

Related Information

Commands: **syspar_ctrl**

emonctrl Script

Purpose

emonctrl – A control script that manages the Emonitor subsystem.

Syntax

emonctrl { –**a** | –**s** | –**k** | –**d** | –**c** | –**t** | –**o** | –**r** | –**h** }

Flags

–**a** Adds the subsystem.

–**s** Starts the subsystem. Not implemented. The subsystem should be started using **Estart -m**

–**k** Stops the subsystem.

–**d** Deletes the subsystem.

–**c** Cleans the subsystems, that is, delete them from all system partitions.

–**t** Turns tracing on for the subsystem. Not used.

–**o** Turns tracing off for the subsystem. Not used.

–**r** Refreshes the subsystem. Not implemented.

–**h** Displays usage information.

Operands

None.

Description

Note: In PSSP 3.1 or later releases, the Emonitor subsystem is no longer needed, since the new Switch Administration daemon and automatic unfence options provide the same functions as the Emonitor subsystem. However, if you turn off the Switch Administration daemon functions you may still want to use the Emonitor subsystem. And if you are using a primary node with a code_version of PSSP 2.4 or earlier in a coexistence environment, the new functions are not supported. You may want to use the Emonitor subsystem in such an environment.

Note: The Emonitor subsystem is not supported on a system with an SP Switch2 switch.

The Emonitor subsystem monitors designated nodes in an attempt to maximize their availability on the switch network.

The **emonctrl** control script controls the operation of the Emonitor subsystem. The subsystem is under the control of the System Resource Controller (SRC) and belongs to a subsystem group called **emon**.

An instance of the Emonitor subsystem can execute on the control workstation for each system partition. Because **Emonitor** provides its services within the scope of a system partition, it is said to be system partition-sensitive. This control script operates in a manner similar to the control scripts of other system partition-sensitive subsystems. It should be issued from the control workstation and is not functional on the nodes.

From an operational point of view, the Emonitor subsystem group is organized as follows:

Subsystem
Emonitor

Subsystem Group
emon

SRC Group
emon

The **emon** group is associated with the **Emonitor** daemon.

On the control workstation, there are multiple instances of **Emonitor**, one for each system partition. Accordingly, the subsystem names on the control workstation have the system partition name appended to them. For example, for system partitions named **sp_prod** and **sp_test**, the subsystems on the control workstation are named **Emonitor.sp_prod** and **Emonitor.sp_test**.

Daemons
Emonitor

The **Emonitor** daemon provides switch node monitoring.

The **emonctrl** script is not normally executed from the command line. It is normally called by the **syspar_ctrl** command during installation of the system, and partitioning or repartitioning of the system.

The **emonctrl** script provides a variety of controls for operating the **Emonitor** daemon:

- Adding, stopping, and deleting the subsystem
- Cleaning up the subsystems, that is, deleting them from all system partitions

Before performing any of these functions, the script obtains the current system partition name and IP address (using the **spget_syspar** command) and the node number (using the **node_number**) command. If the node number is zero, the control script is running on the control workstation. Since the **Emonitor** daemon runs only on the control workstation, the script performs no function when run on a node.

Except for the clean function, all functions are performed within the scope of the current system partition.

Adding the Subsystem

When the –**a** flag is specified, the control script uses the **mkssys** command to add the **Emonitor** daemon to the SRC. The control script operates as follows:

1. It checks whether the Emonitor subsystem already exists in this system partition. If the Emonitor subsystem does exist, it exits.
2. It adds the Emonitor subsystem to the SRC with the system partition name appended.

Starting the Subsystem

This option is unused since the **Emonitor** daemon must be started via **Estart -m**.

Stopping the Subsystem

When the −**k** flag is specified, the control script uses the **stopsrc** command to stop the **Emonitor** daemon in the current system partition.

Deleting the Subsystem

When the −**d** flag is specified, the control script uses the **rmssys** command to remove the Emonitor subsystem from the SRC. The control script operates as follows:

1. It makes sure that the **Emonitor** subsystem is stopped.
2. It removes the **Emonitor** subsystem from the SRC using the **rmssys** command.

Cleaning Up the Subsystems

When the −**c** flag is specified, the control script stops and removes the Emonitor subsystems for all system partitions from the SRC. The control script operates as follows:

1. It stops all instances of subsystems in the subsystem group in all system partitions, using the **stopsrc -g emon** command.
2. It removes all instances of subsystems in the subsystem group in all system partitions from the SRC using the **rmssys** command.

Turning Tracing On

Not currently used.

Turning Tracing Off

Not currently used.

Refreshing the Subsystem

Not currently used.

Logging

While it is running, the **Emonitor** daemon provides information about its operation and errors by writing entries in a log file. The **Emonitor** daemon uses log files called **/var/adm/SPlogs/css/Emonitor.log** and **/var/adm/SPlogs/css/Emonitor.Estart.log**.

Files

/var/adm/SPlogs/css/Emonitor.log
Contains the log of all **Emonitor** daemons on the system.

/var/adm/SPlogs/css/Emonitor.Estart.log
Contains the log of all **Estart** and **Eunfence** commands issued by all **Emonitor** daemons.

Standard Error

This command writes error messages (as necessary) to standard error.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error occurred.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

AIX Version 4 Commands Reference

Information about the System Resource Controller (SRC) in *AIX Version 4 General Programming Concepts: Writing and Debugging Programs*

Location

/usr/lpp/ssp/bin/emonctrl

Related Information

Commands: **Emonitor**, **Estart**, **lssrc**, **startsrc**, **stopsrc**, **syspar_ctrl**

Examples

1. To add the Emonitor subsystem to the SRC in the current system partition, set the **SP_NAME** environment variable to the appropriate system partition name and enter:

```
emonctrl -a
```

2. To stop the Emonitor subsystem in the current system partition, set the **SP_NAME** environment variable to the appropriate system partition name and enter:

```
emonctrl -k
```

3. To delete the Emonitor subsystem from the SRC in the current system partition, set the **SP_NAME** environment variable to the appropriate system partition name and enter:

```
emonctrl -d
```

4. To clean up the Emonitor subsystem on all system partitions, enter:

```
emonctrl -c
```

5. To display the status of all of the subsystems in the Emonitor SRC group, enter:

```
lssrc -g emon
```

6. To display the status of an individual Emonitor subsystem, enter:

   `lssrc -s` *subsystem_name*

7. To display the status of all of the daemons under SRC control, enter:

   ```
   lssrc -a
   ```

Emonitor Daemon

Purpose

Emonitor – Monitors nodes listed in the **/etc/SP/Emonitor.cfg** file in an to attempt to maximize their availability on the switch.

Syntax

Emonitor

Flags

None.

Operands

None.

Description

This command is not valid on a system with an SP Switch2 switch.

Note: In PSSP 3.1 or later releases, the Emonitor subsystem is no longer needed, since the new Switch Administration daemon and automatic unfence options provide the same functions as the Emonitor subsystem. However, if you turn off the Switch Administration daemon functions you may still want to use the Emonitor subsystem. And if you are using a primary node with a code_version of PSSP 2.4 or earlier in a coexistence environment, the new functions are not supported. You may want to use the Emonitor subsystem in such an environment.

Emonitor is a daemon controlled by the System Resource Controller (SRC). It can be used to monitor nodes in a system partition in regard to the their status on the switch. A system-wide configuration file (**/etc/SP/Emonitor.cfg**) lists all nodes on the system to be monitored. The objective is to bring these nodes back up on the switch network when necessary.

Emonitor is invoked with **Estart -m**. Once invoked, it is controlled by SRC so it will restart if it is halted abnormally. If the you decide to end monitoring, you must run **/usr/lpp/ssp/bin/emonctrl -k** to stop the daemon in your system partition.

There is an **Emonitor** daemon for each system partition. The daemon watches for any node coming up (for example, host_responds goes from 0 to 1). When the daemon detects a node coming up, it performs a review of the nodes in the configuration file to check if any node is off the switch network. If any nodes in the specified system partition are off the switch network, it determines a way to bring them back onto the the switch (for example, via **Eunfence** or **Estart**), and takes the appropriate action. To avoid the **Estart** command from being run several times (which can occur if multiple nodes are coming up in sequence), **Emonitor** waits 3 minutes after a node comes up to be sure no other nodes are in the process of coming up. Each time a new node comes up prior to the 3 minute timeout, **Emonitor** resets the timer to a maximum wait of 12 minutes.

Emonitor cannot always bring nodes back on the switch. For example, if any of the following occur:

- The fault service daemon is not operational on the node
- The primary node is down
- There is a hardware error

Problems can occur if the node that is brought off the switch is experiencing a recurring error that causes it to come up and then encounter an error repeatedly. The monitor continually attempts to bring this node into the switch network and could jeopardize the stability of the remaining switch network.

Note: Nodes that will be undergoing hardware or software maintenance should be removed from the **Emonitor.cfg** file during this maintenance to prevent **Emonitor** from attempting to to bring them onto the switch network.

Files

/etc/SP/Emonitor.cfg
Specifies a list of node numbers, one per line, that the user wants monitored by **Emonitor**. This list is system-wide.

Security

You must have root privilege to run this command.

Location

/usr/lpp/ssp/bin

Related Information

Commands: **Eannotator**, **Eclock**, **Efence**, **emonctrl**, **Eprimary**, **Equiesce**, **Estart**, **Etopology**, **Eunfence**, **Eupartition**

enadmin

Purpose

enadmin – Changes the desired state of a specified extension node.

Syntax

enadmin [−**a** {**reset | reconfigure**}] [−**h**] *node_number*

Flags

−a Specifies the desired state to which the extension node is to be set.

reconfigure Once the administrative state of the extension node is placed in this mode, the Simple Network Management Protocol (SNMP) agent managing the extension node will periodically send *trap* messages to the **spmgrd** daemon running on the control workstation requesting configuration data for the extension node. Once the configuration data is received by the agent, it stops sending these requests and uses the configuration data to reconfigure the extension node.

reset Once the administrative state of the extension node is placed in this mode, the SNMP agent managing the extension node will set the extension node to an initial state in which it is no longer an active node on the switch network.

−h Displays usage information.

Operands

node_number
Specifies the node number assigned to the extension node whose state is to be changed.

Description

Use this command to change the administrative state of an extension node. Setting the administrative state of an extension node to **reconfigure** causes configuration data for the extension node to be resent to the extension node's administrative environment. Setting the administrative state of an extension node to **reset** places the extension node in an initial state in which it is no longer active on the switch network.

This command is invoked internally when choosing the **reconfigure** option of the **endefadapter** and **endefnode** commands or the **reset** (−**r**) option of the **enrmnode** command.

You can use the System Management Interface Tool (SMIT) to run this command by selecting the Extension Node Management panel. To use SMIT, enter:

```
smit manage_extnode
```

Standard Output

All informational messages are written to standard output. These messages identify the extension node being changed and indicate when the specified state change has been accepted for processing by the extension node agent (at which point the command is complete). All error messages are also written to standard output.

Exit Values

0 Indicates the administrative state of the extension node was successfully changed.

1 Indicates that an error occurred while processing the command and the administrative state of the extension node was not changed.

Security

You must have root privilege to run this command or be a member of the AIX system group.

Restrictions

This command can only be issued on the control workstation.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP) **ssp.spmgr** file set.

The **spmgrd** SNMP manager daemon on the SP control workstation allows transfer of extension node configuration data from the SP system to an SNMP agent providing administrative support for the extension node. Version 1 of the SNMP protocol is used for communication between the SNMP manager and the SNMP agent. Limited control of an extension node is also possible. An SNMP set-request message containing an object instantiation representing the requested administrative state for the extension node is sent from the SNMP manager to the SNMP agent providing administrative support for the extension node. After the administrative state of an extension node is received by the SNMP agent, the **enadmin** command is completed. Requests for configuration information and information about the state of an extension node are sent to the SNMP manager asynchronously in SNMP trap messages.

Prerequisite Information

IBM RS/6000 SP: Planning, Volume 2, Control Workstation and Software Environment

Location

/usr/lpp/ssp/bin/enadmin

Related Information

Commands: **endefadapter**, **endefnode**, **enrmadapter**, **enrmnode**, **spmgrd**

Examples

1. To request that configuration data for the extension node assigned to node number 9 be sent to its SNMP managing agent, enter:

   ```
   enadmin -a reconfigure 9
   ```

2. To request that the extension node assigned to node number 9 be placed in an initial state and no longer be active on the switch, enter:

   ```
   enadmin -a reset 9
   ```

endefadapter

Purpose

endefadapter – Adds new or changes existing configuration data for an extension node adapter in the System Data Repository (SDR) and optionally performs the reconfiguration request.

Syntax

endefadapter [−**a** *address*] [−**h**] [−**m** *netmask*] [−**r**] *node_number*

Flags

−**a** *address*
: Specifies the IP network address of the extension node adapter. The IP network address must be able to be resolved by the **host** command. This flag is required when adding a new extension node adapter.

−**h**
: Displays usage information.

−**m** *netmask*
: Specifies the netmask for the network on which the extension node adapter resides. This flag is required when adding a new extension node adapter.

−**r**
: Specifies that the extension node adapter will be reconfigured.

Operands

node_number
: Specifies the node number for this extension node adapter. This operand is required.

Description

Use this command to define extension node adapter information in the SDR. The −**a** and −**m** flags and the *node_number* operand are required.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit enter_extadapter
```

Environment Variables

The SP_NAME environment variable is used (if set) to direct this command to a system partition. If the SP_NAME environment variable is not set, the default system partition will be used.

Standard Output

This command writes informational messages to standard output.

Standard Error

This command writes all error messages to standard error.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error occurred and the extension node adapter information was not updated.

Security

You must have root privilege to run this command or be a member of the AIX system group. You must have write access to the SDR to run this command.

Restrictions

This command can only be issued on the control workstation.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP) **ssp.basic** file set.

Prerequisite Information

IBM RS/6000 SP: Planning, Volume 2, Control Workstation and Software Environment

Location

/usr/lpp/ssp/bin/endefadapter

Related Information

Commands: **enadmin**, **endefnode**, **enrmadapter**, **enrmnode**

Examples

1. The following example shows the definition of an extension node adapter for node number 10 with a network address of 129.40.158.137 and a netmask of 255.255.255.0, enter:

```
endefadapter -a 129.40.158.137 -m 255.255.255.0 10
```

2. The following example shows the same definition, but the extension node adapter will be reconfigured after the SDR is updated:

```
endefadapter -a 129.40.158.137 -m 255.255.255.0 -r 10
```

endefnode

Purpose

endefnode – Adds new or changes existing configuration data for an extension node in the System Data Repository (SDR) and optionally performs the reconfiguration request.

Syntax

endefnode [−**a** *hostname*] [−**c** *string*] [−**h**] [−**i** *string*] [−**r**] [−**s** *hostname*] *node_number*

Flags

−**a** *hostname*
: Specifies the administrative host name, which can be resolved to an IP address, associated with the extension nodes's network interface on the administrative network. This flag is required when adding a new extension node.

−**c** *string*
: Specifies the Simple Network Management Protocol (SNMP) community name that the SP SNMP manager and the node's SNMP agent will send in the corresponding field of the SNMP messages. This field consists of 1 to 255 ASCII characters. If the −**c** flag is not specified, the **spmgrd** daemon will use a default SNMP community name. For more information about the default community name, refer to the related extension node publication in the "Related Information" section that follows.

−**h**
: Displays usage information.

−**i** *string*
: Specifies the extension node identifier assigned to the node in its system's administrative environment. This is a text string that uniquely identifies the node to its system. This field consists of 1 to 255 ASCII characters. This flag is required when adding a new extension node.

−**r**
: Specifies that the extension node will be reconfigured.

−**s** *hostname*
: Specifies the host name that can be resolved to an IP address of the extension node's SNMP agent. This flag is required when adding a new extension node.

Operands

node_number
: Specifies the node number for this extension node. The *node_number* specified in this command must be for an unused standard node position that corresponds to the relative node position assigned to the extension node. Otherwise, there would be a conflict in the switch configuration information. This operand is required.

Description

Use this command to define extension node information in the SDR. When adding a new extension node, the −**a**, −**i**, and −**s** flags and the *node_number* operand are required. When changing an existing extension node definition, only the node number is required along with the flag corresponding to the field being changed.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit enter_extnode
```

Environment Variables

The SP_NAME environment variable is used (if set) to direct this command to a system partition. If the SP_NAME environment variable is not set, the default system partition will be used.

Standard Output

This command writes informational messages to standard output.

Standard Error

This command writes all error messages to standard error.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error occurred and the extension node information was not updated.

Security

To run this command you must have root privilege or be a member of the AIX system group. You must also have SDR write access and **hardmon** access to run this command.

Restrictions

This command can only be issued on the control workstation.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP) **ssp.basic** file set.

Prerequisite Information

IBM RS/6000 SP: Planning, Volume 2, Control Workstation and Software Environment

Location

/usr/lpp/ssp/bin/endefnode

Related Information

Commands: **enadmin**, **endefadapter**, **enrmnode**, **enrmadapter**

Refer to the *SP Switch Router Adapter Guide* for information about attaching an IP router extension node to the SP Switch.

Examples

1. The following example shows a definition of an extension node with a node number of 2 that references slot number 13 in a router:

```
endefnode -i 13 -a router1 -s router1 -c spenmgmt 2
```

2. The following example shows a definition of an extension node with a node number of 7 that references slot number 02 in a router. This extension node will also be reconfigured after the SDR is updated.

```
endefnode -i 02 -a grf.pok.ibm.com -s grf.pok.ibm.com -c spenmgmt -r 7
```

enrmadapter

Purpose

enrmadapter – Removes configuration data for an extension node adapter from the System Data Repository (SDR).

Syntax

enrmadapter [−**h**] *node_number*

Flags

−**h** Displays usage information.

Operands

node_number
Specifies the node number for this extension node adapter.

Description

Use this command to remove extension node adapter information from the SDR. The *node_number* operand is required.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit delete_extadapter
```

Environment Variables

The environment variable SP_NAME is used (if set) to direct this command to a system partition. If the SP_NAME environment variable is not set, the default system partition will be used.

Standard Output

This command writes informational messages to standard output.

Standard Error

This command writes all error messages to standard error.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error occurred and the extension node adapter information was not updated.

Security

You must have root privilege to run this command or be a member of the AIX system group. You must have write access to the SDR to run this command.

Restrictions

This command can only be issued on the control workstation.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP) **ssp.basic** file set.

Prerequisite Information

IBM RS/6000 SP: Planning, Volume 2, Control Workstation and Software Environment

Location

/usr/lpp/ssp/bin/enrmadapter

Related Information

Commands: **enadmin**, **endefadapter**, **endefnode**, **enrmnode**

Examples

To remove an extension node adapter with a node number of 12 from the SDR, enter:

```
enrmadapter 12
```

enrmnode

Purpose

enrmnode – Removes configuration data for an extension node in the System Data Repository (SDR).

Syntax

enrmnode [−**h**] [−**r**] *node_number*

Flags

−h Displays usage information.

−r Causes the extension node to be reset.

Operands

node_number
Specifies the node number for this extension node.

Description

Use this command to remove extension node information from the SDR. When removing information, the *node_number* operand is required.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit delete_extnode
```

Environment Variables

The environment variable SP_NAME is used (if set) to direct this command to a system partition. If the SP_NAME environment variable is not set, the default system partition will be used.

Standard Output

This command writes informational messages to standard output.

Standard Error

This command writes all error messages to standard error.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error occurred and the extension node information was not updated.

Security

You must have root privilege to run this command or be a member of the AIX system group. You must have write access to the SDR to run this command.

Restrictions

This command can only be issued on the control workstation.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP) **ssp.basic** file set.

Prerequisite Information

IBM RS/6000 SP: Planning, Volume 2, Control Workstation and Software Environment

Location

/usr/lpp/ssp/bin/enrmnode

Related Information

Commands: **enadmin**, **endefadapter**, **endefnode**, **enrmadapter**

Examples

To remove an extension node with a node number of 2 from the SDR and reset that extension node, enter:

```
enrmnode -r 2
```

Eprimary

Purpose

Eprimary – Assigns or queries the switch primary node and switch primary backup node for a system partition.

Syntax

| **Eprimary** [–**h**] [–**init**] [*node_identifier*] [–**backup** *bnode_identifier*] [–**p** {**0** | **all**}]

Flags

–**h** Displays usage information.

–**init** Initializes or reinitializes the current system partition object. If –**init** is specified without a *node_identifier* or without a *bnode_identifier*, the respective default is used for the primary and primary backup nodes. The lowest numbered node in the system partition is the default primary node, and the furthest node from the primary is the default primary backup node.

When a new system partition object is created, the SDR autounfence attribute is set to "1" (enabled). This attribute determines whether automatic unfence will be enabled or disabled by the Fault Service daemon during switch initialization. Use the **Estart** command to change the value of this attribute.

–**backup** *bnode_identifier*
Specifies the node designated as the oncoming switch primary backup node. It can be a host name, an IP address, a frame,slot pair, or a node number. If a *bnode_identifier* is not specified, the oncoming primary backup node is automatically selected. A dependent node **cannot** be selected as a primary or primary backup node.

| –**p {0 | all}**
| Specifies for which switch plane the operation is to be performed. If not
| specified, the default is to perform the operation for all valid switch planes.
| The only valid switch plane value is 0. This flag is valid only on systems
| with SP Switch2 switches.

Operands

node_identifier
Specifies the node designated as the oncoming switch primary node. It can be a host name, an IP address, a frame,slot pair, or a node number. If a *node_identifier* is not specified, the oncoming primary node is automatically selected. A dependent node **cannot** be selected as a primary or primary backup node.

Note: If no flags or operands are specified, each of the following is displayed:

- Current switch primary node
- Current switch primary backup node
- Oncoming switch primary node
- Oncoming switch primary backup node
- Automatic unfence feature:

  0 - disable automatic unfence

1 - enable automatic unfence

If you are using a primary node with a code version of PSSP 2.4 or earlier in a coexistence environment, the automatic unfence feature is not supported.

Description

Use this command to assign, change, or query the switch primary node or the switch primary backup node. The primary node should not be changed unless the current primary node is becoming unavailable (for example, if the current primary node is to be serviced). The **Estart** command must be issued before a change of the primary node or the primary backup node (using **Eprimary**) takes effect. Also, the **Estart** command must be used if the value of the autounfence attribute needs to be changed.

In an SP Switch network, the primary node takeover facility automatically handles situations (such as a node loss) for each of the primary and primary backup nodes. The primary node replaces a problem primary backup node and the primary backup node automatically takes over for the primary node if the primary node becomes unavailable. Note that the node chosen cannot be a dependent node. The primary backup node should be selected using the following guidelines:

- Specify a node from a different frame other than the primary node, if possible.
- If this node is not selected from a frame that is different from the primary node, it should be a node connected to a switch chip that is different from the primary node's. For example:
  - Node slots 1, 2, 5, and 6 are connected to the same switch chip.
  - Node slots 3, 4, 7, and 8 are connected to the same switch chip.
  - Node slots 9, 10, 13, and 14 are connected to the same switch chip.
  - Node slots 11, 12, 15, and 16 are connected to the same switch chip.

The **Eprimary** command selects a default oncoming primary or oncoming backup primary node if one is not specified. Users receive a warning in the following situations on the oncoming primary or oncoming backup primary nodes:

- If they cannot ping
- If the switch daemon is not up
- If the node is fenced

Security

You must have root privilege to run this command.

Location

/usr/lpp/ssp/bin/Eprimary

Related Information

Commands: **Eannotator**, **Eclock**, **Efence**, **Equiesce**, **Estart**, **Etopology**, **Eunfence**, **Eunpartition**

Examples

1. To query the switch primary and primary backup nodes, enter:

   ```
   Eprimary
   ```

2. To designate an oncoming switch primary node by IP address and let **Eprimary** select an oncoming switch primary backup node, enter:

   ```
   Eprimary 129.33.34.1
   ```

3. To designate an oncoming switch primary node and an oncoming switch primary backup node by IP address, enter:

   ```
   Eprimary 129.33.34.1 -backup 129.33.34.56
   ```

4. To designate an oncoming switch primary node and an oncoming switch primary backup node by host name, enter:

   ```
   Eprimary r11n01 -backup r17n02
   ```

5. To create a system partition object and assign a switch primary backup node by a frame,slot, enter:

   ```
   Eprimary -init 1,2 -backup 1,6
   ```

Equiesce

Purpose

Equiesce – Quiesces the switch by causing the primary and primary backup nodes to shut down switch recovery and primary node takeover.

Syntax

Equiesce [–**h**] [–**p** {**0** | **all**}]

Flags

–**h** Displays usage information.

–**p {0 | all}**
Specifies for which switch plane the operation is to be performed. If not specified, the default is to perform the operation for all valid switch planes. The only valid switch plane value is 0. This flag is valid only on systems with SP Switch2 switches.

Operands

None.

Description

Use this command to disable switch error recovery and primary node takeover. It is used to shut down normal switch error actions when global activities affecting nodes are performed. For example, when all nodes are shutdown or rebooted, they are fenced from the switch by the primary node.

If the primary node is not the first node to shut down during a global shutdown or reboot of the entire system, it may fence all the other nodes including the primary backup node. Primary node takeover can also occur if the primary node is shut down and the backup node remains up. Issuing the **Equiesce** command before the shutdown prevents these situations from occurring.

The **Equiesce** command causes the primary and primary backup nodes to shut down their recovery actions. Data still flows over the switch, but no problems are serviced and primary node takeover is disabled. Only the **Eannotator**, **Eclock**, **Eprimary**, **Estart**, and **Etopology** commands are functional after the **Equiesce** command is issued.

Estart must be issued when the global activity is complete to reestablish switch recovery and primary node takeover.

Note: The Switch Admin Daemon will issue the **Estart** command under certain circumstances, thus reestablishing switch recovery and primary node takeover. To see if you are using the Switch Admin Daemon, issue the following command:

```
lssrc -s swtadmd
```

If the response returned shows that the swtadmd subsystem is active, you may want to turn off the Switch Admin Daemon before issuing the **Equiesce** command. To turn off the Switch Admin Daemon, issue the following command:

```
stopsrc -s swtadmd
```

After issuing the **Estart** command again, you may want to restart the Switch Admin Daemon. To turn on the Switch Admin Daemon, issue the following command:

```
startsrc -s swtadmd
```

Security

You must have root privilege to run this command.

Location

/usr/lpp/ssp/bin/Equiesce

Related Information

Commands: **Eannotator**, **Eclock**, **Efence**, **Eprimary**, **Estart**, **Etopology**, **Eunfence**, **Eunpartition**

Examples

To quiesce the switch before shutting down the system, enter:

```
Equiesce
```

Estart

Purpose

Estart – Starts the switch.

Syntax

Estart [−**h**] [−**m**] [−**autounfence 0 | 1**] [−**p** {**0 | all**}]

Flags

−**h** Displays usage information.

−**m** Specifies that the **Emonitor** daemon should be started. (See **/etc/SP/Emonitor.cfg** for details.) The −**m** flag is valid only on systems with an SP Switch switch.

-autounfence
For SP Switch systems, specifies whether automatic unfence will be enabled (1) or disabled (0). The specified value will be used to update the SDR Switch_partition autounfence attribute. For SP Switch2 systems, specifies whether automatic unfence will be enabled (1) or disabled (0). The specified value will be used to update the SDR Switch_plane autounfence attribute. If this flag is not specified, the current value of the autounfence attribute will be used by the Fault Service daemon on the primary node during switch initialization. Use the **Eprimary** command to display the SDR autounfence attribute.

Note: In PSSP 3.1 or later releases, the Emonitor subsystem is no longer needed, since the new Switch Administration daemon and automatic unfence options provide the same functions as the Emonitor subsystem. However, if you turn off the Switch Administration daemon functions you may still want to use the Emonitor subsystem. And if you are using a primary node with a code_version of PSSP 2.4 or earlier in a coexistence environment, the new functions are not supported. You may want to use the Emonitor subsystem in such an environment.

−**p {0 | all}**
Specifies for which switch plane the operation is to be performed. If not specified, the default is to perform the operation for all valid switch planes. The only valid switch plane value is 0. This flag is valid only on systems with SP Switch2 switches.

Operands

None.

Description

Use this command to start or restart the current system partition based on its switch topology file. (Refer to the **Etopology** command for topology file details.) If the −**m** flag is specified, it will also start the **Emonitor** daemon to monitor nodes on the switch. Refer to the **Emonitor** daemon for additional information. If the **Estart** command is issued when the switch is already running, it causes a switch error, and messages in flight are lost. Applications using reliable protocols on the switch, such as TCP/IP and the MPI User Space library, recover from switch errors. Applications using unreliable protocols on the switch do not recover from switch errors. For this reason, IBM suggests that you be aware of what applications or protocols you are running before you issue the **Estart** command.

If the current primary node is unsuitable for use to start the switch, the current primary backup node is examined. If it is suitable to use as the new current primary node, **Eprimary** is run to make it the new oncoming primary node before starting the switch.

SP Switch Notes:

If you have an SP Switch installed on your system, an oncoming primary node as selected via **Eprimary** is established as primary during **Estart**. If necessary, the topology file is distributed to partition nodes during **Estart**. The topology file to be used is distributed to each of the standard nodes in the system partition via the SP Ethernet:

- If an **/etc/SP/expected.top** file exists on the oncoming primary node
- If the topology file being used is from the System Data Repository (SDR) and not all the nodes in the system partition have the topology file

Otherwise, the topology file is already resident on the nodes and does not need to be distributed.

If automatic unfence is enabled, the autojoin bit for all active nodes will be turned on and the scan will be enabled to check for nodes ready to join the switch. If automatic unfence is disabled, the autojoin bit for all active nodes will be turned off and the Fault Service daemon will not scan for nodes ready to join the switch. For nodes not on the switch during switch initialization, the autojoin bit will remain unchanged. For a fenced node to automatically join the switch during switch initialization, its autojoin bit must be on.

Files

/etc/SP/expected.top.1nsb_8.0isb.0
The standard topology file for systems with a maximum of eight nodes.

/etc/SP/expected.top.1nsb.0isb.0
The standard topology file for one Node Switch Board (NSB) system or a maximum of 16 nodes.

/etc/SP/expected.top.2nsb.0isb.0
The standard topology file for two NSB systems or a maximum of 32 nodes.

/etc/SP/expected.top.3nsb.0isb.0
The standard topology file for three NSB systems or a maximum of 48 nodes.

/etc/SP/expected.top.4nsb.0isb.0
The standard topology file for four NSB systems or a maximum of 64 nodes.

/etc/SP/expected.top.5nsb.0isb.0
The standard topology file for five NSB systems or a maximum of 80 nodes.

/etc/SP/expected.top.5nsb.4isb.0
The standard topology file for five NSB and four Intermediate Switch Board (ISB) systems or a maximum of 80 nodes. This is an advantage-type network with a higher bisectional bandwidth.

/etc/SP/expected.top.6nsb.4isb.0
The standard topology file for six NSB and four ISB systems or a maximum of 96 nodes.

/etc/SP/expected.top.7nsb.4isb.0
The standard topology file for seven NSB and four ISB systems or a maximum of 112 nodes.

/etc/SP/expected.top.8nsb.4isb.0
The standard topology file for eight NSB and four ISB systems or a maximum of 128 nodes.

/etc/SP/expected.top.1nsb_8.0isb.1
The standard topology file for systems with an SP Switch-8 and a maximum of eight nodes.

/etc/SP/Emonitor.cfg
The list of nodes that the user wants monitored via the **Emonitor** daemon (not partition sensitive).

/var/adm/SPlogs/css/dist_topology.log
Contains system error messages if any occurred during the distribution of the topology file to the nodes.

Security

You must have root privilege to run this command.

Location

/usr/lpp/ssp/bin/Estart

Related Information

Commands: **Eannotator**, **Eclock**, **Efence**, **Eprimary**, **Equiesce**, **Etopology**, **Eunfence**, **Eunpartition**

Refer to *IBM RS/6000 SP: Planning, Volume 2, Control Workstation and Software Environment* for details about system partition topology files.

Examples

1. To start the SP Switch, enter:

   ```
   Estart
   ```

2. To start the SP Switch and turn off automatic unfence, enter:

   ```
   Estart -autounfence 0
   ```

Etopology

Purpose

Etopology – Stores or reads a switch topology file into or out of the System Data Repository (SDR).

Syntax

Etopology [−**h**] [−**read**] {*switch_topology_file* | −**d**} [−**p** {**0** | **all**}]

Flags

−h
Displays usage information.

−read
Retrieves the current switch topology file out of the SDR and stores it in the specified *switch_topology_file*. If −**read** is not specified, the specified *switch_topology_file* will be stored in the SDR.

−d
Specifies that the topology filename is to be determined from the contents of the SDR, based on the number and type of switches in the system. On an SP Switch system, this flag is only valid on a system which has not been partitioned.

−p {0 | all}
Specifies for which switch plane the operation is to be performed. If not specified, the default is to perform the operation for all valid switch planes. The only valid switch plane value is 0. This flag is valid only on systems with SP Switch2 switches.

Operands

switch_topology_file
Specifies the full path name of the file into which the current SDR switch topology is to be copied, or the full path name of a switch topology file to store in the SDR. A sequence number is appended to this file name when it is stored in the SDR. This is used to ensure that the appropriate topology file is distributed to the nodes of the system partition.

Description

Use this command to store or retrieve the *switch_topology_file* into or out of the SDR. The switch topology file is used by switch initialization when starting the switch for the current system partition. It is stored in the SDR and can be overridden by having a switch topology file in the **/etc/SP** directory named **expected.top** on the switch primary node.

If you have an SP Switch installed on your system, the current topology file is copied to each node of the subject system partition during an **Estart** and to each targeted node for an **Eunfence**.

Files

/etc/SP/expected.top.1nsb_8.0isb.0
The standard topology file for systems with a maximum of eight nodes.

/etc/SP/expected.top.1nsb.0isb.0
The standard topology file for one Node Switch Board system or a maximum of 16 nodes.

/etc/SP/expected.top.2nsb.0isb.0
The standard topology file for two NSB systems or a maximum of 32 nodes.

/etc/SP/expected.top.3nsb.0isb.0
The standard topology file for three NSB systems or a maximum of 48 nodes.

/etc/SP/expected.top.4nsb.0isb.0
The standard topology file for four NSB systems or a maximum of 64 nodes.

/etc/SP/expected.top.5nsb.0isb.0
The standard topology file for five NSB systems or a maximum of 80 nodes.

/etc/SP/expected.top.5nsb.4isb.0
The standard topology file for five NSB and four Intermediate Switch Board (ISB) systems or a maximum of 80 nodes. This is an advantage-type network with a higher bisectional bandwidth.

/etc/SP/expected.top.6nsb.4isb.0
The standard topology file for six NSB and four ISB systems or a maximum of 96 nodes.

/etc/SP/expected.top.7nsb.4isb.0
The standard topology file for seven NSB and four ISB systems or a maximum of 112 nodes.

/etc/SP/expected.top.8nsb.4isb.0
The standard topology file for eight NSB and four ISB systems or a maximum of 128 nodes.

/etc/SP/expected.top.1nsb_8.0isb.1
The standard topology file for systems with an SP Switch-8 and a maximum of eight nodes.

Security

You must have root privilege to run this command.

Location

/usr/lpp/ssp/bin/Etopology

Related Information

Commands: **Eannotator**, **Eclock**, **Efence**, **Eprimary**, **Equiesce**, **Estart**, **Eunfence**, **Eupartition**

Refer to *IBM RS/6000 SP: Planning, Volume 2, Control Workstation and Software Environment* for information on system partition configurations and topology files.

Examples

1. To store a topology file for a system with up to 96 nodes in the SDR, enter:

   ```
   Etopology /etc/SP/expected.top.6nsb.4isb.0
   ```

2. To store a topology file for a system with up to 16 nodes in the SDR, enter:

   ```
   Etopology /etc/SP/expected.top.1nsb.0isb.0
   ```

3. To retrieve a topology file out of the SDR and store it to a file, enter:

   ```
   Etopology -read /tmp/temporary.top
   ```

Eunfence

Purpose

Eunfence – Adds an SP node to the current *active* switch network that was previously removed from the network.

Syntax

Eunfence [–**h** | [–**G**] [–**p** { **0** | **all**}] *node_specifier* [*node_specifier2*] ...

Flags

–**h**
Displays usage information.

–**G**
Unfences all valid nodes in the list of nodes regardless of system partition boundaries. If the –**G** flag is not used, the **Eunfence** command will only unfence the nodes in the current system partition. All other specified nodes will not be unfenced and a nonzero return code is returned.

–**p {0 | all}**
Specifies for which switch plane the operation is to be performed. If not specified, the default is to perform the operation for all valid switch planes. The only valid switch plane value is 0. This flag is valid only on systems with SP Switch2 switches.

Operands

node_specifier
Specifies a list of nodes that is to rejoin the current switch network. It can be a list of host names, IP addresses, node numbers, frame,slot pairs, or a node group.

Description

Use this command to allow a node to rejoin the current switch network that was previously removed with the **Efence** command.

You can also use this command to allow a node to rejoin the switch network if that node was previously removed from the SP Switch network due to a switch or adapter error.

SP Switch Note:

Eunfence first distributes the current topology file to the nodes before they can be unfenced.

Note: If a host name or IP address is used as the *node_specifier* for a dependent node, it must be a host name or IP address assigned to the adapter that connects the dependent node to the SP Switch. Neither the administrative host name nor the Simple Network Management Protocol (SNMP) agent's host name for a dependent node is guaranteed to be the same as the host name of its switch network interface.

Eunfence attempts to start the fault–service daemon, if it is not currently running, on a node which is to rejoin the current switch network.

Files

/var/adm/SPlogs/css/dist_topology.log
Contains system error messages if any occurred during the distribution of the topology file to the nodes.

Security

You must have root privilege to run this command.

Location

/usr/lpp/ssp/bin/Eunfence

Related Information

Commands: **Eannotator**, **Eclock**, **Efence**, **Eprimary**, **Equiesce**, **Estart**, **Etopology**, **Eunpartition**

Examples

1. To unfence a node by IP address, enter:

   ```
   Eunfence 129.33.34.1
   ```

2. To unfence two nodes by host name, enter:

   ```
   Eunfence r11n01 r11n04
   ```

3. To unfence several nodes by node number, enter:

   ```
   Eunfence 34 43 20 76 40
   ```

4. To unfence node 14 of frame 2 by frame,slots pairs, enter:

   ```
   Eunfence 2,14
   ```

5. If the current system partition has nodes with node numbers 1, 2, 5, and 6 and another system partition has nodes with node numbers 3, 4, 7, and 8, issuing the command:

   ```
   Eunfence 5 6 7 8
   ```

   unfences nodes 5 and 6, but not nodes 7 and 8. As a result, the command returns a nonzero return code.

6. To successfully unfence the nodes in example 5 with the same system partitions, use the **–G** flag as follows:

   ```
   Eunfence -G 5 6 7 8
   ```

Eunpartition

Purpose

Eunpartition – Prepares a system partition for merging with a neighboring system partition.

Syntax

Eunpartition [–**h**]

Flags

–**h** Displays usage information.

If a flag is not specified, **Eunpartition** examines the SP_NAME shell variable and selects a system partition based on its current setting.

Operands

None.

Description

This command is not valid on a system with an SP Switch2 switch.

Use this command to prepare a partitioned configuration for a new system partition definition within an SP cluster.

This command must be executed for each system partition prior to the **spapply_config** command to redefine system partitions.

If you specify **Eunpartition** in error, it will quiesce the primary and primary backup nodes. If this occurs, you must use **Estart** to restart the switch.

Security

You must have root privilege to run this command.

Location

/usr/lpp/ssp/bin/Eunpartition

Related Information

Commands: **Eannotator**, **Eclock**, **Efence**, **Eprimary**, **Equiesce**, **Estart**, **Etopology**, **Eunfence**

Examples

To prepare the current system partition for repartitioning as specified by SP_NAME, enter:

```
Eunpartition
```

export_clients

Purpose

export_clients – Creates or updates the Network File System (NFS) export list for a boot/install server.

Syntax

export_clients [−**h**]

Flags

−**h** Displays usage information. If the command is issued with the −**h** flag, the syntax description is displayed to standard output and no other action is taken.

Operands

None.

Description

Use this command to create or update the NFS export list on a boot/install server node.

Standard Error

This command writes error messages (as necessary) to standard error.

Exit Values

0 Indicates the successful completion of the command.

−**1** Indicates that an error occurred.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Location

/usr/lpp/ssp/bin/export_clients

Related Information

Commands: **setup_server**

Examples

To create or update the NFS export list on a boot/install server node, enter:

```
export_clients
```

ext_srvtab

Purpose

ext_srvtab – Extracts service key files from the Kerberos Version 4 authentication database.

Syntax

ext_srvtab [–**n**] [–**r** *realm*] [*instance ...*]

Flags

–**n** If specified, the master key is obtained from the master key cache file. Otherwise, **ext_srvtab** prompts the user to enter the master key interactively.

–**r** If specified, the realm fields in the extracted file match the given realm rather than the local realm.

Operands

instance Specifies an instance name. On the SP system, service instances consist of the short form of the network names for the hosts on which the service runs.

Description

The **ext_srvtab** command extracts service key files from the Kerberos Version 4 authentication database. The master key is used to extract service key values from the database. For each *instance* specified on the command line, the **ext_srvtab** command creates a new service key file in the current working directory with a file name of *instance*-new-srvtab which contains all the entries in the database with an instance field of *instance*. This new file contains all the keys registered for instances of services defined to run on that host. A user must have read access to the authentication database to execute this command. This command can only be issued on the system on which the authentication database resides.

Files

instance-**new-srvtab**
Service key file generated for *instance*.

/var/kerberos/database/principal.pag, /var/kerberos/database/principal.dir
Files containing the authentication database.

/.k Master key cache file.

Security

You must have root privilege to run this command.

Location

/usr/lpp/ssp/kerberos/etc/ext_srvtab

Related Information

Commands: **kadmin**, **ksrvutil**

Refer to the "RS/6000 SP Files and Other Technical Information" section of *PSSP: Command and Technical Reference* for additional **Kerberos** information.

Examples

If a system has three network interfaces named as follows:

```
ws3e.abc.org

ws3t.abc.org

ws3f.finet.abc.org
```

to re-create the server key file on this workstation (that is an SP authentication server), user root could do the following:

```
# create a new key file in the /tmp  directory for each instance

# Combine the instance files into a single file for the hostname.

# Delete temporary files and protect key file

cd /tmp

/usr/kerberos/etc/ext_srvtab -n ws3e ws3t ws3f

/bin/cat ws3e-new-srvtab ws3t-new-srvtab ws3f-new-srvtab \

   >/etc/krb-srvtab

/bin/rm ws3e-new-srvtab ws3t-new-srvtab ws3f-new-srvtab

/bin/chmod 400 /etc/krb-srvtab
```

fencevsd

Purpose

fencevsd – Prevents an application running on a node or group of nodes from accessing a virtual shared disk or group of virtual shared disks.

Syntax

fencevsd –**v** *vsd_name_list* –**n** *node_list*

Flags

–**v** Specifies one or more virtual shared disk names, separated by commas.

–**n** Specifies one or more node numbers, separated by commas.

Operands

None.

Description

Under some circumstances, the system may believe a node has stopped functioning and begin recovery procedures, when the node is actually operational, but cut off from communication with other nodes running the same application. In this case, the problem node must not be allowed to serve requests for the virtual shared disks it normally serves until recovery is complete and the other nodes running the application recognize the problem node as operational. The **fencevsd** command prevents the problem node from filling requests for its virtual shared disks.

This command can be run from any node where the IBM Recoverable Virtual Shared Disk subsystem is running.

Note: This command will be unsuccessful if you do not specify a current server (primary or backup) to a virtual shared disk with the –**v** flag.

Security

You must be in the AIX **bin** group and have write access to the SDR to run this command.

Prerequisite Information

PSSP: Managing Shared Disks

Location

/usr/lpp/csd/bin/fencevsd

Related Information

Commands: **lsfencevsd**, **lsvsd**, **unfencevsd**, **updatevsdtab**, **vsdchgserver**

Refer to *PSSP: Managing Shared Disks* for information on how to use this command in writing applications.

Examples

To fence the virtual shared disks vsd1 and vsd2 from node 5, enter:

```
fencevsd -v vsd1,vsd2 -n 5
```

get_keyfiles

Purpose

get_keyfiles – Initiates transfer of Kerberos V4 srvtab file from the control workstation to the newly created node for Kerberos V4authentication.

Syntax

get_keyfiles {*keyfile server*}

Flags

None.

Operands

keyfile Specifies the Kerberos V4 srvtab file for the node executing the command.

server Specifies the control workstation which holds the Kerberos V4 srvtab file service keyfiles for all SP nodes.

Description

This program stops all getty processes and removes the "cons" entry from inittab so that it cannot restart until this command is completed. **get_keyfiles** sends a request to the control workstation, and listens on /dev/tty0 for keyfiles. Keyfiles are sent in a uuencoded format over s1term by the control workstation. This program will uudecode the keyfile to its original format and place them in the /spdata/sys1/k4srvtabs directory for the calling program. Once the keyfile transfer is completed, the "cons" entry is added back to inittab, and inittab is refreshed.

Files

The log file **/var/adm/SPlogs/get_keyfiles/get_keyfiles.log** is created.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error or errors occurred.

An unsuccessful run of this command, depending on where an error occurred, will result in the keyfile transfer being unsuccessful.

Security

You must have root privilege to run this command.

Restrictions

This program works with the kfserver function on the control workstation. The kfserver function uses the s1term in write mode. Only one s1term in write mode to a node is allowed at any given time.

Location

/usr/lpp/ssp/bin/get_keyfiles

Examples

To get file c58n01-new-srvtab from c58s.ppd.pok.ibm.com (the control workstation), enter:

```
get_keyfiles c58n01-new-srvtabo c58s.ppd.pok.ibm.com
```

get_vpd

Purpose

get_vpd – Consolidates the Vital Product Data (VPD) files for the nodes and writes the information to a file and optionally to a diskette.

Syntax

get_vpd [–**h**] [–**d**] –**m** *model_number* –**s** *serial_number*

Flags

–**h** Displays usage information.

–**d** Specifies that the Vital Product Data file will be written to a diskette.

–**m** *model_number*
Specifies the machine type model number. The value of the model number is “ MMx”, where MM is the class of the machine:

20 No switch, 2 – 64 nodes
2A No switch, 2 – 8 nodes, 49 inch height
3A 8-port switch, 2 – 8 nodes, 49 inch height
38 8-port switch, 2 – 8 nodes, 79 inch height
30 Single-staged switching, 2 – 80 nodes
40 Dual-staged switching, 62 – 128 nodes

–**s** *serial_number*
Specifies the serial number. The value of the *serial_number* is “pp00sssss”, where:

pp Is 02 for machines built in US (Poughkeepsie) and 51 for machines built in EMEA (Montpelier).

00 Is a mandatory value.

sssss Is the serial number of the machine.

Description

Use this command to consolidate the Vital Product Data (VPD) for the nodes in the RS/6000 SP into a file and to optionally write the file to diskette. The diskette created by this command is sent to IBM manufacturing when an upgrade to the RS/6000 SP hardware is desired. This diskette is used by manufacturing and marketing to configure an upgrade of the RS/6000 SP.

The **get_vpd** command is issued by IBM field personnel to capture VPD information after an upgrade of the system. All installation and configuration of the RS/6000 SP must be complete prior to issuing the **get_vpd** command.

Files

/var/adm/SPlogs/SPconfig/*node_number***.umlc**
Files used as input to this command.

/var/adm/SPlogs/SPconfig/*serial_number***.vpd**
Output file generated by this command.

This command creates the **/var/adm/SPlogs/SPconfig/***serial_number***.vpd** file and optionally writes the file to a diskette.

Standard Error

This command writes all error messages to standard error.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error occurred while processing the VPD information and the command did not complete successfully.

Security

| You must have root privilege and **hardmon** access to run this command.

Restrictions

This command can only be issued on the control workstation.

| The command always generates output in English using the default C locale. It
| ignores the current operating locale.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP) **ssp.basic** file set.

Prerequisite Information

IBM RS/6000 SP: Planning, Volume 2, Control Workstation and Software Environment

Location

/usr/lpp/ssp/install/bin/get_vpd

Examples

1. This example shows the creation of a file containing all of the node VPD information for a model type of 204 and a serial number of 020077650. The output is written to **/var/adm/SPlogs/SPconfig/020077650.vpd**.

   ```
   get_vpd -m 204 -s 020077650
   ```

2. This example shows the creation of a file containing all of the node VPD information for a model type of 306 and a serial number of 510077730. The output is written to **/var/adm/SPlogs/SPconfig/510077730.vpd** and also to diskette.

   ```
   get_vpd -m 306 -s 510077730 -d
   ```

ha_vsd

Purpose

ha_vsd – Starts and restarts the Recoverable Virtual Shared Disk subsystem. This includes configuring virtual shared disks and hashed shared disks as well as activating the recoverability subsystem.

Syntax

ha_vsd [**reset**]

Flags

None.

Operands

reset Stops and restarts the IBM Recoverable Virtual Shared Disk subsystem by stopping the Recoverable Virtual Shared Disk and hc subsystems and then starting them again.

Description

Use this command to start the IBM Recoverable Virtual Shared Disk software after you install it, or, with the **reset** option, to stop and restart the program.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error occurred.

Security

You must have root privilege and write access to the SDR to run this command.

Implementation Specifics

This command is part of the IBM Recoverable Virtual Shared Disk option of PSSP.

Prerequisite Information

See *PSSP: Managing Shared Disks*.

Location

/usr/lpp/csd/bin/ha_vsd

Related Information

Commands: **ha.vsd**, **hc.vsd**

Examples

To stop the Recoverable Virtual Shared Disk subsystem and restart it, enter:

```
ha_vsd reset
```

ha.vsd

Purpose

ha.vsd – Queries and controls the activity of the rvsd daemon of the Recoverable Virtual Shared Disk subsystem.

Syntax

ha.vsd {**adapter_recovery [on** | **off]** | **debug [off]** | **mksrc** | **query** | **quorum** *n* | **qsrc** | **refresh [noquorum]** | **reset** | **reset_quorum** | **rmsrc** | **start** | **stop** | **trace [off]**}

Flags

None.

Operands

adapter_recovery [on | off]
Enables or disables communication adapter recovery. The default is on.

The Recoverable Virtual Shared Disk subsystem must be restarted for this operand to take effect.

debug [off]
Specify **debug** to redirect the Recoverable Virtual Shared Disk subsystem's **stdout** and **stderr** to the console and cause the Recoverable Virtual Shared Disk subsystem to not respawn if it exits with an error. (You can use the **lscons** command to determine the current console.)

The Recoverable Virtual Shared Disk subsystem must be restarted for this operand to take effect.

Once debugging is turned on and the Recoverable Virtual Shared Disk subsystem has been restarted, **ha.vsd trace** should be issued to turn on tracing.

Use this operand under the direction of your IBM service representative.

Note: the default when the node is booted is to have **stdout** and **stderr** routed to the console. If debugging is turned off **stdout** and **stderr** will be routed to **/dev/null** and all further trace messages will be lost. You can determine if debug has been turned on by issuing **ha.vsd qsrc**. If debug has been turned on the return value will be:

```
action = "2"
```

mksrc
Uses **mkssys** to create the Recoverable Virtual Shared Disk subsystem.

query
Displays the current status of the Recoverable Virtual Shared Disk subsystem in detail.

quorum *n*
Sets the value of the quorum, the number of nodes that must be active to direct recovery. Usually, quorum is defined as a majority of the nodes that are defined as virtual shared disk nodes in a system partition, but this command allows you to override that definition. The

Recoverable Virtual Shared Disk subsystem must be in the active state when you issue this command. This is not a persistent change.

qsrc
Displays the System Resource Controller (SRC) configuration of the Recoverable Virtual Shared Disk daemon.

refresh [noquorum]
Uses the **refresh** command to asynchronously start a refresh protocol to all running Recoverable Virtual Shared Disk subsystems. The quorum will be reset before the refresh occurs, unless **noquorum** is specified. Use **ha.vsd query** to check for completion. The following items are refreshed in the device driver:

1. nodes that have been added or deleted
2. vsds and hsds that have been added or deleted
3. changed vsd attributes:
   - option # cache | nocache
   - size_in_MB

reset
Stops and restarts the Recoverable Virtual Shared Disk subsystem.

reset_quorum
Resets the default quorum.

rmsrc
Uses **rmssys** to remove the Recoverable Virtual Shared Disk subsystem.

start
Starts the Recoverable Virtual Shared Disk subsystem.

stop
Stops the Recoverable Virtual Shared Disk subsystem.

trace [off]
Requests or stops tracing of the Recoverable Virtual Shared Disk subsystem. The Recoverable Virtual Shared Disk subsystem must be in the active state when this command is issued.

This operand is only meaningful after the **debug** operand has been used to send **stdout** and **stderr** to the console and the Recoverable Virtual Shared Disk subsystem has been restarted.

Description

Use this command to display information about the Recoverable Virtual Shared Disk subsystem, to change the number of nodes needed for quorum, and to change the status of the subsystem.

You can start the Recoverable Virtual Shared Disk subsystem with the IBM Virtual Shared Disk Perspective. Type `spvsd` and select actions for virtual shared disk nodes.

Exit Values

0
Indicates the successful completion of the command.

nonzero
Indicates that an error occurred.

Security

You must have write access to the SDR to run this command. You must have root privilege to issue the **debug**, **quorum**, **refresh**, **reset**, **start**, **stop**, **trace**, **mksrc**, and **rmsrc** subcommands.

Implementation Specifics

This command is part of the Recoverable Virtual Shared Disk option of PSSP.

Prerequisite Information

See *PSSP: Managing Shared Disks*.

Location

/usr/lpp/csd/bin/ha.vsd

Related Information

Commands: **ha_vsd**, **hc.vsd**

Examples

1. To stop the Recoverable Virtual Shared Disk subsystem and restart it, enter:

   ```
   ha.vsd reset
   ```

   The system returns the messages:

   ```
   Waiting for the rvsd subsystem to exit.

   rvsd subsystem exited successfully.

   Starting rvsd subsystem.

   rvsd subsystem started PID=xxx.
   ```

2. To change the quorum to five nodes of a 16-node SP system, enter:

   ```
   ha.vsd quorum 5
   ```

   The system returns the message:

   ```
   Quorum has been changed from 8 to 5.
   ```

3. To query the rvsd subsystem, enter:

   ```
   ha.vsd query
   ```

   The system displays a message similar to the following:

```
Subsystem         Group            PID     Status

      rvsd             rvsd            18320   active

      rvsd(vsd): quorum= 7, active=1, state=idle, isolation=member,

                 NoNodes=10, lastProtocol=nodes_failing,

                 adapter_recovery=on, adapter_status=up,

                 RefreshProtocol has never been issued from this node,

                 Running function level 3.1.0.0.
```

where:

quorum Is the number of nodes that must join the group before it will be activated.

active Indicates the activation status of the group that is being joined:

- **0:** the group is not active (quorum has not been met).
- **1:** the group is active and the shared disks have been activated.

state Indicates the current protocol that is running.

isolation Indicates the group membership status

- **isolated:** a group "join" has not been proposed.
- **proposed:** a group "join" has been proposed.
- **member:** we are a member (provider) of the group.

NoNodes Indicates the number of nodes that have joined the group

lastProtocol Indicates the last protocol that was run across the group.

adapter_recovery Indicates communication adapter recovery support:

- **on:** adapter recovery is enabled.
- **off:** adapter recovery is disabled.

adapter_status Indicates communication adapter status:

- **up:** the adapter is up.
- **down:** the adapter is down.
- **unknown:** the adapter status is unknown.

RefreshProtocol ... Indicates whether a refresh protocol has been issued from this node. If so, the date and time of success or error will be displayed.

Running function level Indicates the function level that the subsystem is running, in version, release, modification, fix level format (vrmf). (Coexistence with lower levels of the subsystem, may restrict us to running at a reduced function level.)

hacws_verify

Purpose

hacws_verify – Verifies the configuration of both the primary and backup High Availability Control Workstation (HACWS) control workstations.

Syntax

hacws_verify

Flags

None.

Operands

None.

Description

Use this command to verify that the primary and backup control workstations are properly configured to provide HACWS services to the SP system. The **hacws_verify** command inspects both the primary and backup control workstations to verify the following:

- The HACWS software is properly configured
- The resources required to provide control workstation services have been identified to the HACMP software
- The HACWS data stored in the System Data Repository (SDR) is correct
- The HACMP event scripts supplied by HACWS have been identified to the HACMP software

Both the primary and backup control workstations must be running and capable of executing remote commands via the AIX **rsh** command.

The system administrator should run the **hacws_verify** command after HACWS is initially configured. After that, the **hacws_verify** command can be run at any time.

Exit Values

0 Indicates that no problems were found with the HACWS configuration.

nonzero Indicates that problems were found with the HACWS configuration.

Security

Enhanced Security Option

PSSP 3.2 provides the option of running your RS/6000 SP system with an enhanced level of security. This function removes the dependency PSSP has to internally issue **rsh** and **rcp** commands as a root user from a node. When this function is enabled PSSP does not automatically grant authorization for a root user to issue **rsh** and **rcp** commands from a node. If you enable this option some procedures may not work as documented. For example, to run HACMP an administrator must grant the authorizations for a root user to issue **rsh** and **rcp** commands that PSSP would otherwise grant automatically. See the Red Book *Exploiting RS/6000 SP Security: Keeping it Safe* for a description of this function and a complete list of limitations.

Prerequisite Information

Refer to *PSSP: Administration Guide* for additional information on the HACWS option.

Location

/usr/sbin/hacws/hacws_verify

Related Information

PSSP Commands: **install_hacws**, **spcw_addevents**

AIX Commands: **rsh**

Examples

To verify the HACWS configuration, enter:

```
/usr/sbin/hacws/hacws_verify
```

haemcfg

Purpose

haemcfg – Compiles the Event Management objects in the System Data Repository (SDR) and places the compiled information into a binary Event Management Configuration Database (EMCDB) file

Syntax

haemcfg [−**c**] [−**n**]

Flags

−**c** Indicates that you want to check the data in the System Data Repository (SDR) without building the Event Management Configuration Database (EMCDB).

−**n** Indicates that you want to build a test copy of the EMCDB in the current directory.

Operands

None.

Description

The **haemcfg** utility command builds the Event Management Configuration Database (EMCDB) file for a system partition. If no flags are specified, the **haemcfg** command:

- Compiles the Event Management objects in the System Data Repository (SDR)
- Places the compiled information into a binary Event Management Configuration Database (EMCDB) file in a staging directory as **/spdata/sys1/ha/cfg/em.***syspar_name***.cdb**, where *syspar_name* is the system partition name
- Updates the **haem_cdb_version** attribute in the SDR **Syspar** class for the system partition with the current EMCDB version string. The EMCDB version string contains a timestamp and a sequence number.

To place the new EMCDB into production, you must shut down and restart all of this system partition's Event Manager daemons: the daemon on the control workstation and the daemon on each of the system partition's nodes. When the Event Management daemon restarts, it copies the EMCDB from the staging directory to the production directory. The name of the production EMCDB is **/etc/ha/cfg/em.***syspar_name***.cdb**.

If you want to test a new EMCDB, IBM recommends that you create a separate system partition for that purpose.

You must create a distinct EMCDB file for each system partition on the IBM RS/6000 SP. To build an EMCDB file, you must be executing on the control workstation and you must set the SP_NAME environment variable to the appropriate system partition name before you issue the command.

Before you build or replace an EMCDB, it is advisable to issue the **haemcfg** command with the debugging flags.

The −**c** flag lets you check the validity of the Event Management data that resides in the SDR. This data was previously loaded through the **haemloadcfg** command. If any of the data is not valid, the command writes an error message that describes the error.

When the −**c** flag is processed, the command validates the data in the SDR, but does not create a new EMCDB file and does not update the EMCDB version string in the SDR.

The −**n** flag lets you build a test EMCDB file in the current directory. If anything goes wrong with the creation of the new file, the command writes an error message that describes the error.

When the −**n** flag is processed, the command uses the data in the SDR to create a test EMCDB file in the current directory, but it does not update the EMCDB version string in the SDR. If any of the data in the SDR is not valid, the command stops at the first error encountered.

If you specify both flags on the command line, the **haemcfg** command performs the actions of the −**c** flag.

After you have checked the data and the build process, issue the **haemcfg** command without any flags. This builds the new EMCDB file, places it in the **/spdata/sys1/ha/cfg** directory, and updates the EMCDB version string in the SDR.

Files

/spdata/sys1/ha/cfg/em.*syspar_name***.cdb**
Contains the most recently compiled EMCDB file for the system partition specified by *syspar_name*. This file will be placed into production when all of the Event Management daemons in the system partition are next restarted.

/etc/ha/cfg/em.*syspar_name***.cdb**
Contains the production EMCDB file for the system partition specified by *syspar_name*. This EMCDB file is currently in use by the Event Management subsystem.

Standard Output

When the command executes successfully, it writes the following informational messages:

Reading Event Management data for partition *syspar_name*

CDB=*new_EMCDB_file_name* Version=*EMCDB_version_string*

Standard Error

This command writes error messages (as necessary) to standard error.

Errors can result from causes that include:

- Internal space allocation errors

- SDR access errors
- Errors that occur while trying to access system partition data in the SDR
- Event Management data in the SDR is not valid
- EMCDB file access errors
- Insufficient user authorization for the command
- Trying to update the EMCDB in the **/spdata/sys1/ha/cfg** directory from a node of the system partition, rather than from the control workstation.

For a listing of the errors that the **haemcfg** command can produce, see *PSSP: Message Reference*.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error occurred. It is accompanied by one or more error messages that indicate the cause of the error.

Security

You must have write access to the SDR to run this command.

To place an EMCDB file for a system partition into the **/spdata/sys1/ha/cfg** directory, you must be running with an effective user ID of **root** on the control workstation. Before running this command, you must set the SP_NAME environment variable to the appropriate system partition name.

Restrictions

To place an EMCDB file for a system partition into the **/spdata/sys1/ha/cfg** directory, you must be running with an effective user ID of **root** on the control workstation. Before running this command, you must set the SP_NAME environment variable to the appropriate system partition name.

If you run the **haemcfg** command without any flags, the command stops at the first error it encounters. With the **-c** flag on, the command continues, letting you obtain as much debugging information as possible in one pass. To reduce your debugging time, therefore, run the command with the debugging flags first.

Implementation Specifics

This command is part of RS/6000 Cluster Technology (RSCT), which is included with the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

For a general overview of configuring Event Management, see “The Event Management Subsystem” chapter of *PSSP: Administration Guide*.

For a description of the SDR classes and attributes that are related to the EMCDB, see *IBM RS/6000 Cluster Technology: Event Management Programming Guide and Reference*.

Location

/usr/sbin/rsct/bin/haemcfg

Related Information

Commands: **haemloadcfg**

Examples

1. To validate the Event Management data in the System Data Repository (without creating a new EMCDB file), enter:

   ```
   haemcfg -c
   ```

   If there are any errors in the data, the command writes appropriate error messages.

   To fix the errors, replace the data in the SDR. For more information, see the man page for the **haemloadcfg** command.

2. To create a test EMCDB file in the current directory, enter:

   ```
   haemcfg -n
   ```

   If there are any problems in creating the file, the command writes appropriate error messages.

3. To compile a new EMCDB file for a system partition from the Event Management data that resides in the SDR and place it into the staging directory:

   a. Make sure you are executing with root authority on the control workstation.

   b. Make sure that the SP_NAME environment variable is set to the name of the appropriate system partition.

   c. Enter:

      ```
      haemcfg
      ```

      In response, the command creates a new EMCDB file, places it in the staging directory as **/spdata/sys1/ha/cfg/em.** *syspar_name***.cdb**, where *syspar_name* is the name of the current system partition, and updates the EMCDB version string in the SDR.

haemctrl Script

Purpose

haemctrl – A control script that starts the Event Management subsystem.

Syntax

haemctrl {–**a** | –**s** | –**k** | –**d** | –**c** | –**u** | –**t** | –**o** | –**r** | –**h**}

Flags

–**a** Adds the subsystem.

–**s** Starts the subsystem.

–**k** Stops the subsystem.

–**d** Deletes the subsystem.

–**c** Cleans the subsystems, that is, deletes them from all system partitions.

–**u** Unconfigures the subsystems from all system partitions.

–**t** Turns tracing on for the subsystem.

–**o** Turns tracing off for the subsystem.

–**r** Refreshes the subsystem.

–**h** Displays usage information.

Operands

None.

Description

Event Management is a distributed subsystem of RSCT that provides a set of high availability services for the IBM RS/6000 SP. By matching information about the state of system resources with information about resource conditions that are of interest to client programs, it creates events. Client programs can use events to detect and recover from system problems, thus enhancing the availability of the SP system.

The **haemctrl** control script controls the operation of the Event Management subsystem. The subsystem is under the control of the System Resource Controller (SRC) and belongs to a subsystem group called **haem**. Associated with each subsystem is a daemon.

The **haemctrl** script also controls the operation of the AIX Resource Monitor subsystem. The subsystem is under SRC control and also belongs to the **haem** subsystem group. Associated with each subsystem is a daemon.

Instances of the Event Management and AIX Resource Monitor subsystems execute on the control workstation and on every node of a system partition. Because Event Management provides its services within the scope of a system partition, these subsystems are said to be system partition-sensitive. This control script operates in a manner similar to the control scripts of other system partition-sensitive subsystems. It can be issued from either the control workstation or any of the system partition's nodes.

From an operational point of view, the Event Management subsystem group is organized as follows:

Subsystem
: Event Management

Subsystem Group
: **haem**

SRC Subsystem
: **haem**

The **haem** subsystem is associated with the **haemd** daemon.

The subsystem name on the nodes is **haem**. There is one of each subsystem per node and it is associated with the system partition to which the node belongs.

On the control workstation, there are multiple instances of each subsystem, one for each system partition. Accordingly, the subsystem names on the control workstation have the system partition name appended to them. For example, for system partitions named **sp_prod** and **sp_test**, the subsystems on the control workstation are named **haem.sp_prod** and **haem.sp_test**.

haemaixos

The **haemaixos** subsystem is associated with the **harmad** daemon.

The subsystem name on the nodes is **haemaixos**. There is one of each subsystem per node and it is associated with the system partition to which the node belongs.

On the control workstation, there are multiple instances of each subsystem, one for each system partition. Accordingly, the subsystem names on the control workstation have the system partition name appended to them. For example, for system partitions named **sp_prod** and **sp_test**, the subsystems on the control workstation are named **haemaixos.sp_prod** and **haemaixos.sp_test**.

Daemons
: **haemd** and **harmad**

The **haemd** daemon provides the Event Management services. The **harmad** daemon is the resource monitor for AIX operating system resources.

The **haemctrl** script is not normally executed from the command line. It is normally called by the **syspar_ctrl** command during installation of the system, and partitioning or repartitioning of the system.

The **haemctrl** script provides a variety of controls for operating the Event Management subsystem:

- Adding, starting, stopping, and deleting the subsystem
- Cleaning up the subsystems, that is, deleting them from all system partitions
- Unconfiguring the subsystems from all system partitions

- Turning tracing on and off
- Refreshing the Event Management subsystem

Before performing any of these functions, the script obtains the current system partition name and IP address (using the **spget_syspar** command) and the node number (using the **node_number**) command. If the node number is zero, the control script is running on the control workstation.

Except for the clean and unconfigure functions, all functions are performed within the scope of the current system partition.

Adding the Subsystem

When the **-a** flag is specified, the control script uses the **mkssys** command to add the Event Management and AIX Resource Monitor subsystems to the SRC. The control script operates as follows:

1. It makes sure that the **haem** and **haemaixos** subsystems are stopped.
2. It gets the port number for the **haem** subsystem for this system partition from the **Syspar_ports** class of the System Data Repository (SDR) and ensures that the port number is set in the **/etc/services** file. If there is no port number in the SDR and this script is running on the control workstation, the script obtains a port number. If the script is running on a node and there is no port number in the SDR, the script ends with an error. The range of valid port numbers is 10000 to 10100, inclusive.

   The service name that is entered in the **/etc/services** file is **haem.***syspar_name*.
3. It removes the **haem** and **haemaixos** subsystems from the SRC (just in case they are still there).
4. It adds the **haem** subsystem to the SRC. On the control workstation, the IP address of the system partition is specified to be supplied as an argument to the daemon by the **mkssys** command.
5. It adds the **haemaixos** subsystem to the SRC.
6. It adds an entry for the **haem** group to the **/etc/inittab** file. The entry ensures that the group is started during boot. However, if **haemctrl** is running on a High Availability Control Workstation (HACWS), no entry is made in the **/etc/inittab** file. Instead, HACWS manages starting and stopping the group.
7. The **haemrm** group is added using the **mkgroup** command, if it does not already exist. Any errors that occur are written to a log file named **/var/ha/log/em.mkgroup**.
8. The **/var/ha/lck/haem** and **/var/ha/soc/haem** directories are created, if they don't already exist. Any errors that occur are written to a log file named **/var/ha/log/em.mkdir**.
9. On the control workstation, it creates the Event Management Configuration Database (EMCDB). First, it runs the **haemloadcfg** command to load the SDR with the Event Management configuration data that is contained in the **haemloadlist** file. Then, it runs the **haemcfg** command to compile the data in the SDR and create the binary Event Management Configuration Database. Any errors that occur are written to a log file named **/var/ha/log/em.loadcfg.***syspar_name*.

For more information about configuring Event Management data, see the *IBM RS/6000 Cluster Technology: Event Management Programming Guide and Reference*.

Then it gets the port number for the subsystem from the **SP_ports** class of the System Data Repository (SDR) and ensures that the port number is set in the **/etc/services** file. This port number is used for remote connections to Event Management daemons that are running on the control workstation. If there is no port number in the SDR, the script obtains one and sets it in the **/etc/services** file. The range of valid port numbers is 10000 to 10100, inclusive.

The service name is **haemd**.

Starting the Subsystem

When the −**s** flag is specified, the control script uses the **startsrc** command to start the Event Management subsystem, **haem**, and the AIX Resource Monitor subsystem, **haemaixos**.

Stopping the Subsystem

When the −**k** flag is specified, the control script uses the **stopsrc** command to stop the Event Management subsystem, **haem**, and the AIX Resource Monitor subsystem, **haemaixos**..

Deleting the Subsystem

When the −**d** flag is specified, the control script uses the **rmssys** command to remove the Event Management and AIX Resource Monitor subsystems from the SRC. The control script operates as follows:

1. It makes sure that the **haem** and **haemaixos** subsystems are stopped.
2. It removes the **haem** and **haemaixos** subsystems from the SRC using the **rmssys** command.
3. It removes the port number from the **/etc/services** file.
4. If there are no other subsystems remaining in the **haem** group, it removes the entry for the **haem** group from the **/etc/inittab** file.

Cleaning Up the Subsystems

When the −**c** flag is specified, the control script stops and removes the Event Management subsystems for all system partitions from the SRC. The control script operates as follows:

1. It stops all instances of subsystems in the subsystem group in all partitions, using the **stopsrc -g haem** command.
2. It removes the entry for the **haem** group from the **/etc/inittab** file.
3. It removes all instances of subsystems in the subsystem group in all partitions from the SRC using the **rmssys** command.
4. It removes all Event Management entries from the **/etc/services** file. These include the port numbers for the subsystems as well as the port number used for remote connections.

Unconfiguring the Subsystems

When the −**u** flag is specified, the control script performs the function of the −**c** flag in all system partitions and then removes all port numbers from the SDR allocated by the Event Management subsystems.

Note: The −**u** flag is effective only on the control workstation.

Prior to executing the **haemctrl** command with the −**u** flag on the control workstation, the **haemctrl** command with the −**c** flag must be executed from all of the nodes. If this subsystem is not successfully cleaned from all of the nodes, different port numbers may be used by this subsystem, leading to undefined behavior.

Turning Tracing On

When the −**t** flag is specified, the control script turns tracing on for the **haemd** daemon, using the **haemtrcon** command. Tracing for the **harmad** daemon is also enabled, using the **traceson** command.

Turning Tracing Off

When the −**o** flag is specified, the control script turns tracing off for the **haemd** daemon, using the **haemtrcoff** command. Tracing for the **harmad** daemon is also disabled, using the **tracesoff** command.

Refreshing the Subsystem

When the −**r** flag is specified, the control script refreshes the subsystem using the **refresh** command. This results in the Event Management subsystem attempting to use the current SP Trusted Services authentication methods. Note that this command only initiates the refresh operation. Use the **lssrc** −**ls haem.***syspar_name* command on the control workstation or the **lssrc** −**ls haem** command on a node to determine the current security state of the Event Management subsystem within the system partition. See the Event Management chapter in the PSSP: Administration Guide for further information.

Logging

While it is running, the Event Management daemon normally provides information about its operation and errors by writing entries to the AIX error log. If it cannot, errors are written to a log file called **/var/ha/log/em.default.***syspar_name*.

Files

/var/ha/log/em.default.*syspar_name*
: Contains the default log of the **haemd** daemon on the system partition named *syspar_name*.

/var/ha/log/em.loadcfg.*syspar_name*
: Contains a log of any errors that occurred while creating the Event Management Configuration Database for the system partition named *syspar_name* using the **haemcfg** command.

/var/ha/log/em.trace.*syspar_name*
: Contains the trace log of the **haemd** daemon on the system partition named *syspar_name*.

/var/ha/log/em.mkgroup
Contains a log of any errors that occurred while creating the **haemrm** group.

/var/ha/log/em.mkdir
Contains a log of any errors that occurred while creating the **/var/ha/lck/haem** and **/var/ha/soc/haem** directories.

Standard Error

This command writes error messages (as necessary) to standard error.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error occurred.

Security

You must have root privilege and write access to the SDR to run this command.

Implementation Specifics

This command is part of RS/6000 Cluster Technology (RSCT), which is included with the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

"The Event Management Subsystem" chapter of *PSSP: Administration Guide*

IBM RS/6000 Cluster Technology: Event Management Programming Guide and Reference

AIX Version 4 Commands Reference

Information about the System Resource Controller (SRC) in *AIX Version 4 General Programming Concepts: Writing and Debugging Programs*

Location

/usr/sbin/rsct/bin/haemctrl

Related Information

Commands: **haemcfg**, **haemd**, **haemloadcfg**, **haemtrcoff**, **haemtrcon**, **lssrc**, **startsrc**, **stopsrc**, **syspar_ctrl**

Examples

1. To add the Event Management subsystem to the SRC in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

```
haemctrl -a
```

2. To start the Event Management subsystem in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   haemctrl -s
   ```

3. To stop the Event Management subsystem in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   haemctrl -k
   ```

4. To delete the Event Management subsystem from the SRC in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   haemctrl -d
   ```

5. To clean up the Event Management subsystem on all system partitions, enter:

   ```
   haemctrl -c
   ```

6. To unconfigure the Event Management subsystem from all system partitions, on the control workstation, enter:

   ```
   haemctrl -u
   ```

7. To turn tracing on for the Event Management daemon in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   haemctrl -t
   ```

8. To turn tracing off for the Event Management daemon in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   haemctrl -o
   ```

9. To display the status of all of the subsystems in the Event Management SRC group, enter:

   ```
   lssrc -g haem
   ```

10. To display the status of an individual Event Management subsystem on a node, enter:

    ```
    lssrc -s haem
    ```

    To display the status of an individual Event Management subsystem on the control workstation, enter:

    ```
    lssrc -s haem.syspar_name
    ```

    where *syspar_name* is the system partition name.

11. To display detailed status about an individual Event Management subsystem on a node, enter:

    ```
    lssrc -l -s haem
    ```

    To display detailed status about an individual Event Management subsystem on the control workstation, enter:

    ```
    lssrc -l -s haem.syspar_name
    ```

    where *syspar_name* is the system partition name.

In response, the system returns information that includes the running status of the subsystem, the settings of trace flags, the version number of the Event Management Configuration Database, the time the subsystem was started, the connection status to Group Services and peer Event Management subsystem, and the connection status to Event Management clients, if any.

12. To display the status of all of the daemons under SRC control, enter:

```
lssrc -a
```

haemd Daemon

Purpose

haemd – The Event Manager daemon, which observes resource variable instances that are updated by Resource Monitors and generates and reports events to client programs.

Syntax

haemd

Flags

No specifiable flags.

Operands

No specifiable operands.

Description

The **haemd** daemon is the Event Manager daemon. The daemon observes resource variable instances that are updated by Resource Monitors and generates and reports events to client programs.

One instance of the **haemd** daemon executes on the control workstation for each system partition. An instance of the **haemd** daemon also executes on every node of a system partition. The **haemd** daemon is under System Resource Controller (SRC) control.

Because the daemon is under SRC control, it cannot be started directly from the command line. It is normally started by the **haemctrl** command, which is in turn called by the **syspar_ctrl** command during installation of the system, and partitioning or repartitioning of the system. If you must start or stop the daemon directly, use the **haemctrl** command.

When SRC creates the **haemd** daemon, the actual program started is **haemd_SP**. The **haemd_SP** program, after collecting information needed by the daemon, then executes the **haemd** program. In other words, the **haemd_SP** program is replaced by the **haemd** program in the process created by SRC.

For more information about the Event Manager daemon, see the **haemctrl** man page.

Implementation Specifics

This command is part of RS/6000 Cluster Technology (RSCT), which is included with the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

"The Event Management Subsystem" chapter of *PSSP: Administration Guide*

IBM RS/6000 Cluster Technology: Event Management Programming Guide and Reference

AIX Version 4 Commands Reference

Information about the System Resource Controller (SRC) in *AIX Version 4 General Programming Concepts: Writing and Debugging Programs*

Location

/usr/sbin/rsct/bin/haemd

Related Information

Commands: **haemctrl** and **haemd_SP**

Examples

See the **haemctrl** command.

haemd_SP

Purpose

haemd_SP – Start-up program for the Event Manager daemon.

Syntax

haemd_SP [**-T** *group_name*] [**-d** *trace_arg*] ... [*syspar_IPaddr*]

Flags

-T *group_name* Indicates that the Event Manager daemon is to execute in test mode. The daemon joins a peer group using the group name specified by *group_name* and uses a local copy of the CDB.

-d *trace_arg* Enables tracing for the daemon activity specified by *trace_arg*. This flag may be specified multiple times.

Operands

syspar_IPaddr Specifies the IP address of the system partition in which the **haemd** daemon is to execute. If the daemon is executing on the control workstation, this argument must be specified. Otherwise, the argument is ignored, if present.

Description

The **haemd_SP** program is the start-up program for the **haemd** daemon. When the Event Management subsystem is configured in the System Resource Controller (SRC) by the **haemctrl** command, **haemd_SP** is specified as the program to be started. The *syspar_IPaddr* argument is configured if necessary.

This program can only be invoked by the SRC. To start the Event Management subsystem use the **haemctrl** command.

The **-d** flag should only be used under the direction of the IBM Support Center. The possible trace arguments are the same as for the **haemtrcon** command, except for **regs** and **dinsts**. To use this flag the **haem** subsystem definition (**haem.***syspar_name* on the control workstation) in the SRC must be changed using the **chssys** command with the **-a** argument. Then the daemon must be stopped and then restarted.

The **-T** flag can be used when testing a new resource monitor. Refer to Coding and Testing the Resource Monitor, Alternative Testing Method in Chapter 1 of the *IBM RS/6000 Cluster Technology: Event Management Programming Guide and Reference*.

Implementation Specifics

This command is part of RS/6000 Cluster Technology (RSCT), which is included with the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

The "Event Management Subsystem" chapter of *PSSP: Administration Guide*

IBM RS/6000 Cluster Technology: Event Management Programming Guide and Reference

AIX Version 4 Commands Reference

Information about the System Resource Controller (SRC) in *AIX Version 4 General Programming Concepts: Writing and Debugging Programs*

Location

/usr/sbin/rsct/bin/haemd_SP

Related Information

Commands: **haemctrl**, **haemd**, **haemtrcon**

Examples

See the **haemctrl** command.

haemloadcfg

Purpose

haemloadcfg – Loads Event Management configuration data into the System Data Repository (SDR).

Syntax

haemloadcfg [−**d**] [−**r**] *loadlist_file*

Flags

−d Deletes objects from the SDR that match objects in the load list file.

−r Replaces objects in the SDR by matching objects in the load list file. Any unmatched objects in the load list file are added to the SDR.

Operands

loadlist_file The name of the file that contains the Event Management configuration data to be loaded into the SDR. To load the default PSSP configuration data, specify **/usr/lpp/ssp/install/config/haemloadlist**.

Description

The **haemloadcfg** utility command loads Event Management configuration data into the SDR. Note that before you invoke **haemloadcfg**, you must ensure that the SP_NAME environment variable is set to the appropriate system partition name.

The configuration data is contained in a load list file, whose format is described by the man page for the **haemloadlist** file. For details on the SDR classes and attributes that you can use to specify Event Management configuration data, see *IBM RS/6000 Cluster Technology: Event Management Programming Guide and Reference*.

To load the default Event Management configuration data for PSSP, specify the load list file as **/usr/sbin/rsct/install/config/haemloadlist**.

To add Event Management configuration data for other Resource Monitors, create a file in load list format and specify its name on the command.

Without any flags, the **haemloadcfg** command does not replace existing objects in the SDR. The data in the load list file is matched with the existing objects in the SDR based on key attributes, as follows:

SDR Class	Key Attributes
EM_Resource_Variable	**rvName**
EM_Resource_ID	**riResource_name, riElement_name**
EM_Structured_Byte_String	**sbsVariable_name**, **sbsField_name**
EM_Resource_Class	**rcClass**
EM_Resource_Monitor	**rmName**

Note that the way in which the **haemloadcfg** command handles existing SDR objects is different from the way in which the **SDRCreateObjects** command handles them. The **SDRCreateObjects** command creates a new object as long as the attributes, taken as a group, are unique.

To change a nonkey attribute of an Event Management object that already exists in the SDR, change the attribute in the load list file. Then run the **haemloadcfg** command using the **−r** flag and the name of the load list file. All objects in the SDR are replaced by matching objects in the load list file using the key attributes to match. Any unmatched objects in the load list file are added to the SDR.

To delete Event Management objects from the SDR, create a load list file with the objects to be deleted. Only the key attributes need to be specified. Then run the **haemloadcfg** command using the **−d** flag and the name of the load list file. All objects in the SDR that match objects in the load list file are deleted. No unmatched objects, if any in the load list file, are added to the SDR.

Under any circumstances, duplicate objects in the load list file, based on matches in key attributes, are ignored. However, such duplicate objects are written to standard output.

This release of RS/6000 Cluster Technology has changed (from PSSP release 2.4) several names in the SDR Event Management configuration data:

Table 3.

Old Class Name	New Class Name
EM_Instance_Vector	**EM_Resource_ID**

Table 4.

Old Attribute Name	New Attribute Name
ivResource_name	**riResource_name**
ivElement_name	**riElement_name**
ivElement_description	**riElement_description**
rvPredicate	**rvExpression**
rvIndex_vector	**rvIndex_element**

If there is configuration data present in the SDR from a prior release, the **haemloadcfg** command automatically migrates the data from the old names to the new names the first time the command is executed. After successful migration the objects in the **EM_Instance_Vector** class are deleted.

Note that **rvExpression** and **rvIndex_element** are added to the definition of the **EM_Resource_Variable** class; **rvPredicate** and **rvIndex_vector** are still present in this class but are no longer used after migration.

For compatibility the **haemloadcfg** command accepts load list files using the old class and attribute names.

Files

/usr/sbin/rsct/install/config/haemloadlist
Contains the default configuration data for the Event Management subsystem.

Standard Error

This command writes error messages (as necessary) to standard error.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error occurred. It is accompanied by one or more error messages that indicate the cause of the error.

Security

You must have root privilege and write access to the SDR to run this command.

You should be running on the control workstation. Before running this command, you must set the SP_NAME environment variable to the appropriate system partition name.

Implementation Specifics

This command is part of RS/6000 Cluster Technology (RSCT), which is included with the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

For a general overview of configuring Event Management, see “The Event Management Subsystem” chapter of *PSSP: Administration Guide*.

For details on the System Data Repository classes and attributes for Event Management configuration Database, see *IBM RS/6000 Cluster Technology: Event Management Programming Guide and Reference*.

Location

/usr/sbin/rsct/install/bin/haemloadcfg

Related Information

Commands: **haemcfg**, **SDRCreateObjects**, **SDRDeleteObjects**

Files: **haemloadlist**

Also, for a description of the SDR classes for Event Management configuration data, see *IBM RS/6000 Cluster Technology: Event Management Programming Guide and Reference*.

Examples

1. To load PSSP's default Event Management configuration data into the SDR, enter:

   ```
   haemloadcfg /usr/sbin/rsct/install/config/haemloadlist
   ```

2. To load Event Management configuration data for a new Resource Monitor that is contained in a file called **/usr/local/config/newrmloadlist**, enter:

   ```
   haemloadcfg /usr/local/config/newrmloadlist
   ```

   If nonkey attributes in this load list file are later changed, update the SDR by entering:

   ```
   haemloadcfg -r /usr/local/config/newrmloadlist
   ```

   If this new Resource Monitor is no longer needed, its configuration data is removed from the SDR by entering:

   ```
   haemloadcfg -d /usr/local/config/newrmloadlist
   ```

haemqvar

Purpose

haemqvar– Queries resource variables.

Syntax

haemqvar [**-H** *domain* | **-S** *domain*] [**-c** | **-d** | **-i**] [**-f** *file*] [**-h**] [*class var rsrcID* [**...**]]

Flags

–H *domain* Queries resource variables in the HACMP domain specified by *domain*.

–S *domain* Queries resource variables in the SP domain specified by *domain*.

–c Queries current resource variable values.

–d Queries resource variable definitions but produces short form output.

–i Queries instances of resource variables.

–f *file* Queries resource variables specified in *file*.

–h Displays a usage statement only.

Operands

class Specifies the name of the resource variable class or a null string.

var Specifies the name of the resource variable or a null string.

rsrcID Specifies a resource ID or an asterisk.

Description

The **haemqvar** command queries the Event Management subsystem for information about resource variables. By default, the command writes to standard output the definitions for all resource variables in the current SP domain. That is, the current SP system partition as defined by the **SP_NAME** environment variable. If **SP_NAME** is not set the default system partition is used. The **-S** flag can be used to specify another SP domain (system partition). To query variables in an HACMP domain, use the **-H** flag. For an SP domain, the *domain* flag argument is a system partition name. For a HACMP domain, the *domain* flag argument is a HACMP cluster name. When the **-H** flag is specified, the command must be executed on one of the nodes in the HACMP cluster.

The following information is reported for each resource variable definition:

- Variable Name
- Value Type
- Data Type
- SBS Format (if data type is Structured Byte String)
- Initial Value
- Class
- Locator

- Variable Description
- Resource ID and its description
- Default Expression (if defined) and its description

Since the default behavior of this command can produce a large amount of output, standard output should be redirected to a file.

If the **-d** flag is specified only the resource variable name and a short description are written to standard output, one name and description per line.

If the **-c** flag is specified the current values of all resource variables instances are written to standard output, one per line. The line of output contains the location of the resource variable instance (node number), the resource variable name, the resource ID of the instance and the resource variable instance value. If the resource variable is a Structured Byte String (SBS) data type, then the value of each SBS field is reported.

The **-i** flag reports the same information as the **-c** flag except that the value of the variable instance is the last known value rather than the current value. The **-i** flag is useful for determining what resource variable instances exist.

For both the **-c** and the **-i** flags, if an error is encountered in obtaining information about a resource variable instance, the output line contains an error message, symbolic error codes, the location of where the error originated (if it can be determined), the resource variable name and the resource ID.

To return information about specific resource variables, specify the *class*, *var* and *rsrcID* operands. These operands can be repeated to specify additional resource variables. In addition, the *var* and *rsrcID* operands can be wildcarded to match a number of resource variables. Note that null string operands or an asterisk must be quoted in the shells.

If *class* is not a null string, then all variables in the specified class, as further limited by the *var* and *rsrcID* arguments, are targets of the query. If *class* is a null string, then variables of all classes, as further limited by the *var* and *rsrcID* arguments, are targets of the query.

The *var* argument can be wildcarded in one of two ways:

- Specify the variable name as a null string
- Truncate the name after any component

When the resource variable name is wildcarded in the first manner, then all resource variables, as further limited by the *class* and *rsrcID* arguments, are targets of the query. When the resource variable name is wildcarded in the second manner, all resource variables whose high-order (leftmost) components match the *var* argument, as further limited by the *class* and *rsrcID* arguments, are targets of the query.

All resource variable instances (or definitions if neither the **-c** nor the **-i** flags are specified) of the variables specified by the *class* and *var* arguments that match the *rsrcID* argument are the targets of the query.

If neither the **-c** nor the **-i** flags are specified, the *rsrcID* argument is a semicolon-separated list of resource ID element names. If either the **-c** or the **-i** flags are specified, the *rsrcID* argument is a semicolon-separated list of name/value pairs. A name/value pair consists of a resource ID element name followed by an equal sign followed by a value of the resource ID element. An element value may consist of a single value, a range of values, a comma-separated list of single values or a comma-separated list of ranges. A range takes the form *a-b* and is valid only for resource ID elements of type integer (the type information can be obtained from the variable definition). There can be no blanks in the resource ID.

A resource ID element is wildcarded by specifying its value as the asterisk character. Only variables that are defined to contain the elements, and only the elements, specified in the *rsrcID* argument are targets of the query. If any element of the resource ID consists of the asterisk character, rather than a name/value pair (or just a name if querying for definitions), all variables that are defined to contain *at least* the remaining specified elements are targets of the query. The entire resource ID is wildcarded if it consists of only the asterisk character; all instances of all resource variables, as further limited by the *class* and *var* arguments, are targets of the query.

Note that the *rsrcID* argument must be quoted in the shells if it contains semicolons or asterisks.

The *class*, *var* and *rsrcID* operands can be placed in a file, one set of operands per line, instead of being specified as command arguments. Use the **-f** flag to specify the name of the file to the command. If the **-f** flag is used, any operands to the command are ignored. Within the file, null strings are specified as two adjacent double quote characters and a completely wildcarded resource ID can either be a single asterisk or a double quoted asterisk ("*"). On each line the arguments must be separated by white space (blanks or tabs).

Following are some examples of using wildcards in the **rsrcID** argument:

```
NodeNum=5;VG=rootvg;LV=hd4

NodeNum=*;VG=rootvg;LV=hd4

NodeNum=*;VG=*;LV=*

NodeNum=9

NodeNum=*

NodeNum=9;VG=*;*

NodeNum=*;*
```

For these examples, assume the **class** and **var** arguments are null strings. If either the **class** or **var** arguments or both are not null strings, targets for the query are restricted accordingly.

In the first three examples, all variables whose resource IDs are defined to contain the elements **NodeNum**, **VG** and **LV**, and only those elements, are matched. In the first example, only one instance is matched. In the second example, one instance from each node is matched. In the third example, all instances of the matching resource variables are matched.

In the fourth example, all variables whose resource IDs are defined to contain only the element **NodeNum** are matched. The instances matched are associated with node 9. In the fifth example, the same set of variables are matched, but all instances of each variable are matched.

In the sixth example, all variables whose resource IDs are defined to contain elements **NodeNum** and **VG**, as well as zero or more additional elements, are matched. The instances matched are associated with node 9. In the last example, all variables whose resource IDs are defined to contain the element **NodeNum**, as well as zero or more additional elements, are matched. All instances of the variables are matched.

Given the flexibility in specifying resource variables for query, it is possible that no resource variable instance or resource variable definition will match. If there is no match appropriate error information is reported, either in the form described above or as follows. If the specification of the *class*, *var* or *rsrcID* arguments are in error, the output line contains an error message, symbolic error codes and the specified class name, resource variable name and resource ID.

|
Security

| You must have Event Manager access to run this command. See *PSSP:*
| *Administration Guide* for more information.

Implementation Specifics

This command is part of RS/6000 Cluster Technology (RSCT), which is included with the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Location

/usr/sbin/rsct/bin/haemqvar

Related Information

IBM RS/6000 Cluster Technology: Event Management Programming Guide and Reference

| The Event Manager chapter in *PSSP: Administration Guide*

Examples

1. To obtain the definitions of all resource variables in the current SP system partition and place the output in a file, enter:

   ```
   haemqvar > vardefs.out
   ```

2. To obtain a short form list of all resource variables whose resource IDs contain the element **VG**, in the HACMP cluster named HAcluster, enter:

   ```
   haemqvar -H HAcluster -d "" "" "VG;*"
   ```

   To obtain resource variables whose resource IDs contain only the elements **VG** and **NodeNum**, enter:

   ```
   haemqvar -H HAcluster -d "" "" "VG;NodeNum"
   ```

3. To obtain the amount of **/tmp** file system space used in all nodes in the current SP domain, enter:

```
haemqvar -c "" IBM.PSSP.aixos.FS.%totused "VG=rootvg;LV=hd3;*"
```

haemtrcoff

Purpose

haemtrcoff – Turns tracing off for the Event Manager daemon.

Syntax

haemtrcoff –**s** *subsys_name* –**a** *trace_list*

Flags

–**s** *subsys_name*
: Specifies the name of the Event Management subsystem. On a node of a system partition, this is **haem**. On the control workstation, this is **haem.***syspar_name*, where *syspar_name* is the name of the system partition for which you want to specify the subsystem. This argument must be specified.

–**a** *trace_list*
: Specifies a list of trace arguments. Each argument specifies the type of activity for which tracing is to be turned off. At least one argument must be specified. If more than one argument is specified, the arguments must be separated by commas. The list may not include blanks.

Operands

The following trace arguments may be specified:

init Stops tracing the initialization of the Event Manager daemon.

config Stops dumping information from the configuration file.

insts Stops tracing resource variable instances that are handled by the daemon.

rmctrl Stops tracing Resource Monitor control.

cci Stops tracing the client communication (internal) interface.

emp Stops tracing the event manager protocol.

obsv Stops tracing resource variable observations.

evgn Stops tracing event generation and notification.

reg Stops tracing event registration and unregistration.

pci Stops tracing the peer communication (internal) interface.

msgs Stops tracing all messages that come to and are issued from the daemon.

query Stops tracing queries that are handled by the daemon.

gsi Stops tracing the Group Services (internal) interface.

eval Stops tracing expression evaluation.

rdi Stops tracing the reliable daemon (internal) interface.

bli Stops tracing the back level (internal) interface, used for handling nodes that are running a level of PSSP that is earlier than PSSP 2.2.

sched Stops tracing the internal scheduler.

shm Stops tracing shared memory management activity.

all Stops tracing all activities.

all_but_msgs
Stops tracing all activities except for messages. Message activity is defined by the **msgs** argument.

Description

The **haemtrcoff** command is used to turn tracing off for specified activities of the Event Manager daemon. Trace output is placed in an Event Management trace log for the system partition.

Use this command only under the direction of the IBM Support Center. It provides information for debugging purposes and may degrade the performance of the Event Management subsystem or anything else that is running in the system partition. Do **not** use this command during normal operation.

Files

/var/ha/log/em.trace.*syspar_name*
Contains the trace log of the **haemd** daemon on the system partition named *syspar_name*.

/var/ha/log/em.msgtrace.*syspar_name*
Contains message trace output from the Event Manager daemon on the system partition named *syspar_name*.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of RS/6000 Cluster Technology (RSCT), which is included with the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

"The Event Management Subsystem" chapter of *PSSP: Administration Guide*

Location

/usr/sbin/rsct/bin/haemtrcoff

Related Information

Commands: **haemctrl**, **haemd**, **haemtrcon**

Examples

In the following examples, the SP system has two system partitions named **sp_prod** and **sp_test**. The instances of the Event Management subsystem on the control workstation of the SP are named **haem.sp_prod** and **haem.sp_test**, respectively. The instance of the Event Management subsystem that runs on any node of either system partition is named **haem**.

1. To turn off all tracing for the Event Management subsystem on the control workstation for the **sp_prod** system partition, login to the control workstation and enter:

   ```
   haemtrcoff -s haem.sp_prod -a all
   ```

2. To turn off all tracing for the Event Management subsystem on one of the nodes of the **sp_test** system partition, login to the node and enter:

   ```
   haemtrcoff -s haem -a all
   ```

3. To turn off all tracing of initialization and configuration for the Event Management subsystem on the control workstation for the **sp_test** system partition, login to the control workstation and enter:

   ```
   haemtrcoff -s haem.sp_test -a init,config
   ```

haemtrcon

Purpose

haemtrcon – Turns tracing on for the Event Manager daemon.

Syntax

haemtrcon –**s** *subsys_name* –**a** *trace_list*

Flags

–**s** *subsys_name*
Specifies the name of the Event Management subsystem. On a node of a system partition, this is **haem**. On the control workstation, this is **haem.***syspar_name*, where *syspar_name* is the name of the system partition for which you want to specify the subsystem. This argument must be specified.

–**a** *trace_list*
Specifies a list of trace arguments. Each argument specifies the type of activity for which tracing is to be turned on. At least one argument must be specified. If more than one argument is specified, the arguments must be separated by commas. The list may not include blanks.

Operands

The following trace arguments may be specified:

init Traces the initialization of the Event Manager daemon.

config Dumps information from the configuration file.

insts Traces resource variable instances that are handled by the daemon.

rmctrl Traces Resource Monitor control.

cci Traces the client communication (internal) interface.

emp Traces the event manager protocol.

obsv Traces resource variable observations.

evgn Traces event generation and notification.

reg Traces event registration and unregistration.

pci Traces the peer communication (internal) interface.

msgs Traces all messages that come to and are issued from the daemon.

query Traces queries that are handled by the daemon.

gsi Traces the Group Services (internal) interface.

eval Traces expression evaluation.

rdi Traces the reliable daemon (internal) interface.

bli Traces the back level (internal) interface, used for handling nodes that are running a level of PSSP that is earlier than PSSP 2.2.

sched Traces the internal scheduler.

shm Traces shared memory management activity.

all Traces all activities.

all_but_msgs
Traces all activities except for messages. Message activity is defined by the **msgs** argument.

regs Traces currently registered events.

dinsts Traces all resource variable instances known to the daemon.

iolists Traces immediate observation lists

olists Traces observation lists

Description

The **haemtrcon** command is used to turn tracing on for specified activities of the Event Manager daemon. Trace output is placed in an Event Management trace log for the system partition. When used, the **regs**, **dinsts**, **iolists**, and **olists** arguments perform a one-time trace. The specified information is placed in the trace log, but no further tracing is done.

Use this command only under the direction of the IBM Support Center. It provides information for debugging purposes and may degrade the performance of the Event Management subsystem or anything else that is running in the system partition. Do **not** use this command to turn tracing on during normal operation.

Files

/var/ha/log/em.trace.*syspar_name*
Contains the trace log of the **haemd** daemon on the system partition named *syspar_name*.

/var/ha/log/em.msgtrace.*syspar_name*
Contains message trace output from the Event Manager daemon on the system partition named *syspar_name*.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of RS/6000 Cluster Technology (RSCT), which is included with the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

"The Event Management Subsystem" chapter of *PSSP: Administration Guide*

Location

/usr/sbin/rsct/bin/haemtrcon

Related Information

Commands: **haemctrl**, **haemd**, **haemtrcoff**

Examples

In the following examples, the SP system has two system partitions named **sp_prod** and **sp_test**. The instances of the Event Management subsystem on the control workstation of the SP are named **haem.sp_prod** and **haem.sp_test**, respectively. The instance of the Event Management subsystem that runs on any node of either system partition is named **haem**.

1. To turn on all tracing for the Event Management subsystem on the control workstation for the **sp_prod** system partition, login to the control workstation and enter:

   ```
   haemtrcon -s haem.sp_prod -a all
   ```

2. To turn on all tracing for the Event Management subsystem on one of the nodes of the **sp_test** system partition, login to the node and enter:

   ```
   haemtrcon -s haem -a all
   ```

3. To turn on all tracing of initialization and configuration for the Event Management subsystem on the control workstation for the **sp_test** system partition, login to the control workstation and enter:

   ```
   haemtrcon -s haem.sp_test -a init,config
   ```

haemunlkrm

Purpose

haemunlkrm – Unlocks and starts a Resource Monitor.

Syntax

haemunlkrm –**s** *subsys_name* –**a** *resmon_name*

Flags

–**s** *subsys_name*
Specifies the name of the Event Management subsystem. On a node of a system partition, this is **haem**. On the control workstation, this is **haem.***syspar_name*, where *syspar_name* is the name of the system partition for which you want to specify the subsystem. This argument must be specified.

–**a** *resmon_name*
Specifies the name of the Resource Monitor to unlock and start.

Description

If the Event Management daemon cannot successfully start a resource monitor after three attempts within a two hour interval, or if the daemon has successfully connected to the instances of a resource monitor **N** times within a two hour interval, the resource monitor is "locked" and no further attempts are made to start it or to connect to any of its instances. **N** is three times the maximum number of instances a resource monitor can have, as specified by the **rmNum_instances** attribute in the **EM_Resource_Monitor** SDR class. Once the cause of the problem is determined and the problem corrected, the **haemunlkrm** command can be used to unlock the Resource Monitor and attempt to start it or connect to the resource monitor instances.

The status of the Event Manager daemon, as displayed by the **lssrc** command, indicates if a Resource Monitor is locked.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of RS/6000 Cluster Technology (RSCT), which is included with the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

"The Event Management Subsystem" chapter of *PSSP: Administration Guide*

Location

/usr/sbin/rsct/bin/haemunlkrm

Examples

If the output of the **lssrc** command indicates that the hardware Resource Monitor IBM.PSSP.hmrmd is locked, then after correcting the condition that prevented the Resource Monitor from being started, enter:

```
haemunlkrm -s haem -a IBM.PSSP.hmrmd
```

Note: This example applies to unlocking a Resource Monitor on a node.

hagsctrl Script

Purpose

hagsctrl – A control script that starts the Group Services subsystems.

Syntax

hagsctrl {–**a** | –**s** | –**k** | –**d** | –**c** | –**u** | –**t** | –**o** | –**r** | –**h**}

Flags

–**a** Adds the subsystems.

–**s** Starts the subsystems.

–**k** Stops the subsystems.

–**d** Deletes the subsystems.

–**c** Cleans the subsystems, that is, delete them from all system partitions.

–**u** Unconfigures the subsystems from all system partitions.

–**t** Turns tracing on for the subsystems.

–**o** Turns tracing off for the subsystems.

–**r** Refreshes the subsystem.

–**h** Displays usage information.

Operands

None.

Description

Group Services provides distributed coordination and synchronization services for other distributed subsystems running on a set of nodes on the IBM RS/6000 SP. The **hagsctrl** control script controls the operation of the subsystems that are required for Group Services. These subsystems are under the control of the System Resource Controller (SRC) and belong to a subsystem group called **hags**. Associated with each subsystem is a daemon.

An instance of the Group Services subsystem executes on the control workstation and on every node of a system partition. Because Group Services provides its services within the scope of a system partition, its subsystems are said to be system partition-sensitive. This control script operates in a manner similar to the control scripts of other system partition-sensitive subsystems. It can be issued from either the control workstation or any of the system partition's nodes.

From an operational point of view, the Group Services subsystem group is organized as follows:

Subsystem Group Services

Subsystem Group **hags**

SRC Subsystems	**hags** and **hagsglsm** The **hags** subsystem is associated with the **hagsd** daemon. The **hagsglsm** subsystem is associated with the **hagsglsmd** daemon. The subsystem names on the nodes are **hags** and **hagsglsm**. There is one of each subsystem per node and it is associated with the system partition to which the node belongs. On the control workstation, there are multiple instances of each subsystem, one for each system partition. Accordingly, the subsystem names on the control workstation have the system partition name appended to them. For example, for system partitions named **sp_prod** and **sp_test**, the subsystems on the control workstation are named **hags.sp_prod**, **hags.sp_test**, **hagsglsm.sp_prod**, and **hagsglsm.sp_test**.
Daemons	**hagsd** and **hagsglsmd** The **hagsd** daemon provides the majority of the Group Services functions. The **hagsglsmd** daemon provides global synchronization services for the switch adapter membership group.

The **hagsctrl** script is not normally executed from the command line. It is normally called by the **syspar_ctrl** command during installation of the system, and partitioning or repartitioning of the system.

The **hagsctrl** script provides a variety of controls for operating the Group Services subsystems:

- Adding, starting, stopping, and deleting the subsystems
- Cleaning up the subsystems, that is, deleting them from all system partitions
- Unconfiguring the subsystems from all system partitions
- Turning tracing on and off

Before performing any of these functions, the script obtains the current system partition name (using the **spget_syspar** command) and the node number (using the **node_number**) command. If the node number is zero, the control script is running on the control workstation.

Except for the clean and unconfigure functions, all functions are performed within the scope of the current system partition.

Adding the Subsystem

When the **–a** flag is specified, the control script uses the **mkssys** command to add the Group Services subsystems to the SRC. The control script operates as follows:

1. It makes sure that both the **hags** and **hagsglsm** subsystems are stopped.
2. It gets the port number for the **hags** subsystem for this system partition from the **Syspar_ports** class of the System Data Repository (SDR) and ensures that the port number is set in the **/etc/services** file. If there is no port number in the

SDR and this script is running on the control workstation, the script obtains a port number. If the script is running on a node and there is no port number in the SDR, the script ends with an error. The range of valid port numbers is 10000 to 10100, inclusive.

The service name that is entered in the **/etc/services** file is **hags.***syspar_name*.

3. It removes the **hags** and **hagsglsm** subsystems from the SRC (just in case they are still there).
4. It adds the **hags** and **hagsglsm** subsystems to the SRC. The system partition name is configured as a daemon parameter on the **mkssys** command.
5. It adds an entry for the **hags** group to the **/etc/inittab** file. The entry ensures that the group is started during boot. However, if **hagsctrl** is running on a High Availability Control Workstation (HACWS), no entry is made in the **/etc/inittab** file. Instead, HACWS manages starting and stopping the group.

Starting the Subsystem

When the **−s** flag is specified, the control script uses the **startsrc** command to start the Group Services subsystems, **hags** and **hagsglsm**.

Stopping the Subsystem

When the **−k** flag is specified, the control script uses the **stopsrc** command to stop the Group Services subsystems, **hags** and **hagsglsm**.

Deleting the Subsystem

When the **−d** flag is specified, the control script uses the **rmssys** command to remove the Group Services subsystems from the SRC. The control script operates as follows:

1. It makes sure that both the **hags** and **hagsglsm** subsystems are stopped.
2. It removes the **hags** and **hagsglsm** subsystems from the SRC using the **rmssys** command.
3. It removes the port number from the **/etc/services** file.
4. If there are no other subsystems remaining in the **hags** group, it removes the entry for the **hags** group from the **/etc/inittab** file.

Cleaning Up the Subsystems

When the **−c** flag is specified, the control script stops and removes the Group Services subsystems for all system partitions from the SRC. The control script operates as follows:

1. It stops all instances of subsystems in the subsystem group in all partitions, using the **stopsrc -g hags** command.
2. It removes the entry for the **hags** group from the **/etc/inittab** file.
3. It removes all instances of subsystems in the subsystem group in all partitions from the SRC using the **rmssys** command.

Unconfiguring the Subsystems

When the −**u** flag is specified, the control script performs the function of the −**c** flag in all system partitions and then removes all port numbers from the SDR allocated by the Group Services subsystems.

Note: The −**u** flag is effective only on the control workstation.

Prior to executing the **hagsctrl** command with the −**u** flag on the control workstation, the **hagsctrl** command with the −**c** flag must be executed from all of the nodes. If this subsystem is not successfully cleaned from all of the nodes, different port numbers may be used by this subsystem, leading to undefined behavior.

Turning Tracing On

When the −**t** flag is specified, the control script turns tracing on for the **hagsd** daemon, using the **traceson** command. Tracing is not available for the **hagsglsmd** daemon.

Turning Tracing Off

When the −**o** flag is specified, the control script turns tracing off (returns it to its default level) for the **hagsd** daemon, using the **tracesoff** command. Tracing is not available for the **hagsglsmd** daemon.

Refreshing the Subsystem

The −**r** flag has no effect for this subsystem.

Logging

While they are running, the Group Services daemons provide information about their operation and errors by writing entries in a log file in the **/var/ha/log** directory.

Each daemon limits the log size to a pre-established number of lines (by default, 5,000 lines). When the limit is reached, the daemon appends the string **.bak** to the name of the current log file and begins a new log. If a **.bak** version already exists, it is removed before the current log is renamed.

Files

/var/ha/log/hags_*nodenum_instnum*. *syspar_name*
: Contains the log of the **hagsd** daemons on the nodes.

/var/ha/log/hags.*syspar_name_nodenum_instnum***.***syspar_name*
: Contains the log of each **hagsd** daemon on the control workstation.

/var/ha/log/hagsglsm_*nodenum_instnum***.***syspar_name*
: Contains the log of the **hagsglsmd** daemons on the nodes.

/var/ha/log/hagsglsm.*syspar_name_nodenum_instnum***.***syspar_name*
: Contains the log of each **hagsglsmd** daemon on the control workstation.

The file names include the following variables:

- *nodenum* is the node number on which the daemon is running
- *instnum* is the instance number of the daemon

- *syspar_name* is the name of the system partition in which the daemon is running.

Standard Error

This command writes error messages (as necessary) to standard error.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error occurred.

Security

| You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

"The Group Services Subsystem" chapter of *PSSP: Administration Guide*

PSSP: Group Services Programming Guide and Reference

AIX Version 4 Commands Reference

Information about the System Resource Controller (SRC) in *AIX Version 4 General Programming Concepts: Writing and Debugging Programs*

Location

/usr/sbin/rsct/bin/hagsctrl

Related Information

Commands: **hagsd**, **hagsglsmd**, **lssrc**, **startsrc**, **stopsrc**, **syspar_ctrl**

Examples

1. To add the Group Services subsystems to the SRC in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

```
hagsctrl -a
```

2. To start the Group Services subsystems in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

```
hagsctrl -s
```

3. To stop the Group Services subsystems in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

```
hagsctrl -k
```

4. To delete the Group Services subsystems from the SRC in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   hagsctrl -d
   ```

5. To clean up the Group Services subsystems on all system partitions, enter:

   ```
   hagsctrl -c
   ```

6. To unconfigure the Group Services subsystem from all system partitions, on the control workstation, enter:

   ```
   hagsctrl -u
   ```

7. To turn tracing on for the Group Services daemon in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   hagsctrl -t
   ```

8. To turn tracing off for the Group Services daemon in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   hagsctrl -o
   ```

9. To display the status of all of the subsystems in the Group Services SRC group, enter:

   ```
   lssrc -g hags
   ```

10. To display the status of an individual Group Services subsystem, enter:

    lssrc -s *subsystem_name*

11. To display detailed status about an individual Group Services subsystem, enter:

    lssrc -l -s *subsystem_name*

    In response, the system returns information that includes the running status of the subsystem, the number and identity of connected GS clients, information about the Group Services domain, and the number of providers and subscribers in established groups.

12. To display the status of all of the daemons under SRC control, enter:

    ```
    lssrc -a
    ```

hagsd Daemon

Purpose

hagsd – A Group Services daemon that provides a general purpose facility for coordinating and monitoring changes to the state of an application that is running on a set of nodes.

Syntax

hagsd *daemon_name*

Flags

None.

Operands

daemon_name Specifies the name used by the daemon to name log files and identify its messages in the error log.

Description

The **hagsd** daemon is part of the Group Services subsystem, which provides a general purpose facility for coordinating and monitoring changes to the state of an application that is running on a set of nodes. This daemon provides most of the services of the subsystem.

One instance of the **hagsd** daemon executes on the control workstation for each system partition. An instance of the **hagsd** daemon also executes on every node of a system partition. The **hagsd** daemon is under System Resource Controller (SRC) control.

Because the daemon is under SRC control, it is better not to start it directly from the command line. It is normally called by the **hagsctrl** command, which is in turn called by the **syspar_ctrl** command during installation of the system, and partitioning or repartitioning of the system. If you must start or stop the daemon directly, use the **startsrc** or **stopsrc** command.

For more information about the Group Services daemons, see the **hagsctrl** man page.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

"The Group Services Subsystem" chapter of *PSSP: Administration Guide*

PSSP: Group Services Programming Guide and Reference

AIX Version 4 Commands Reference

Information about the System Resource Controller (SRC) in *AIX Version 4 General Programming Concepts: Writing* *and Debugging Programs*

Location

/usr/sbin/rsct/bin/hagsd

Related Information

Commands: **hagsctrl**, **hagsglsmd**

Examples

See the **hagsctrl** command.

hagsglsmd Daemon

Purpose

hagsglsmd – A Group Services daemon that provides global synchronization services for the switch adapter membership group.

Syntax

hagsglsmd *daemon_name*

Flags

None.

Operands

daemon_name Specifies the name used by the daemon to name log files and identify its messages in the error log.

Description

The **hagsglsmd** daemon is part of the Group Services subsystem, which provides a general purpose facility for coordinating and monitoring changes to the state of an application that is running on a set of nodes.

One instance of the **hagsglsmd** daemon executes on the control workstation for each system partition. An instance of the **hagsglsmd** daemon also executes on every node of a system partition. The **hagsglsmd** daemon is under System Resource Controller (SRC) control.

Because the daemon is under SRC control, it is better not to start it directly from the command line. It is normally called by the **hagsctrl** command, which is in turn called by the **syspar_ctrl** command during installation of the system, and partitioning or repartitioning of the system. If you must start or stop the daemon directly, use the **startsrc** or **stopsrc** command.

For more information about the Group Services daemons, see the **hagsctrl** man page.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

"The Group Services Subsystem" chapter of *PSSP: Group Services Programming Guide and Reference*

PSSP: Group Services Programming Guide and Reference

AIX Version 4 Commands Reference

Information about the System Resource Controller (SRC) in *AIX Version 4 General Programming Concepts: Writing and Debugging Programs*

Location

/usr/sbin/rsct/bin/hagsglsmd

Related Information

Commands: **hagsctrl**, **hagsd**

Examples

See the **hagsctrl** command.

hardmon Daemon

Purpose

hardmon – Monitors and controls the state of the SP hardware.

Syntax

hardmon [–**B**] [–**r** *poll_rate*] [–**d** *debug_flag*] ...

Flags

–**B** Executes the daemon in diagnostic mode.

–**r** *poll_rate*
Specifies the rate, in seconds, at which the daemon polls each frame for state information.

–**d** *debug_flag*
Specifies the daemon debug flag to be set in the daemon. Refer to the **hmadm** command for possible values of *debug_flag*. Multiple –**d** debug flags can be specified.

Operands

None.

Description

hardmon is the Hardware Monitor daemon. The daemon monitors and controls the state of the SP hardware contained in one or more SP frames. This command is not normally executed from the command line. Access to the Hardware Monitor is provided by the **hmmon**, **hmcmds**, **spmon**, **s1term**, and **nodecond** commands. Control of the Hardware Monitor daemon is provided by the **hmadm** command. These commands are the Hardware Monitor "client" commands.

The Hardware Monitor daemon executes on the Monitor and Control Node (MACN). The MACN is that IBM RS/6000 workstation to which the RS-232 lines are connected to the frames. The MACN is one and the same as the control workstation. The daemon is managed by the System Resource Controller (SRC). When the MACN is booted, an entry in **/etc/inittab** invokes the **startsrc** command to start the daemon. The daemon is configured in the SRC to be restarted automatically if it terminates for any reason other than the **stopsrc** command. The SRC subsystem name for the Hardware Monitor daemon is **hardmon**.

hardmon obtains configuration information from the System Data Repository (SDR). The SP_ports object class specifies the port number that the daemon is to use to accept TCP/IP connections from the client commands. The port number is obtained from the object whose *daemon* attribute value matches **hardmon** and whose *host_name* attribute value matches the host name of the workstation on which the daemon is executing. There must be one **hardmon** object in SP_ports for the MACN. The Frame object class contains an object for each frame in the SP system.

The attributes of interest to the daemon are **frame_number**, **tty**, and **MACN**. When started, the daemon fetches all those objects in the Frame class whose *MACN* attribute value matches the host name of the workstation on which the daemon is executing. For each frame discovered in this manner, the daemon saves the frame number and opens the corresponding tty device. When all frames have been configured, the daemon begins to poll the frames for state information. Current state and changed state can then be obtained using the **hmmon** and **spmon** commands. The **hmcmds** and **spmon** commands can be used to control the hardware within the frames.

The daemon also reads the file **/spdata/sys1/spmon/hmthresholds** for values used to check boundary conditions for certain state variables. Refer to the **/spdata/sys1/spmon/hmthresholds** man page for more information. Finally, the **/spdata/sys1/spmon/hmacls** file is read for Access Control List (ACL) information. Refer to the **hmadm** command and the **/spdata/sys1/spmon/hmacls** file for more information on ACLs.

All errors detected by the Hardware Monitor daemon are written to the AIX error log.

The flags in the SRC subsystem object for the **hardmon** subsystem should not normally be changed. For example, if the poll rate is more than 5 seconds, the **nodecond** command can have unpredictable results. Upon request from IBM support for more information to aid in problem determination, debug flags can be set using the **hmadm** command.

If the High Availability Control Workstation (HACWS) Frame Supervisor (type 20) or the SEPBU HACWS Frame Supervisor (type 22) is installed in the SP frames, the –**B** flag is used to run the Hardware Monitor daemon in diagnostic mode. This diagnostic mode is used to validate that the frame ID written into the Supervisor matches the frame ID configured in the SDR for that frame. Normally, the frame ID is automatically written into the Supervisor during system installation. The frame ID is written into the frame to detect cabling problems in an HACWS configuration. In a non-HACWS SP configuration, the –**B** flag is useful whenever the RS232 cables between the frames and MACN are changed (but only if one or more frames contain a type 20 or type 22 supervisor). The **hardmon** command can be executed directly from the command line with the –**B** flag, but only after the currently running daemon is stopped using the **stopsrc** command. Diagnostic messages are written to the AIX error log. The daemon exits when all frames are validated.

Frame ID validation is also performed every time the daemon is started by the System Resource Controller. Any frame that has a frame ID mismatch can be monitored, but any control commands to the frame are ignored until the condition is corrected. A frame with a mismatch is noted in the System Monitor Graphical User Interface as well as in the AIX error log. The **hmcmds** command can be used to set the currently configured frame ID into a type 20 or type 22 supervisor after it is verified that the frame is correctly connected to the MACN.

Additional Configuration Information: The Hardware Monitor subsystem also obtains information from the system partition and the Syspar_map object classes in the SDR. While this information is not used by the **hardmon** daemon itself, it is used by the **hardmon** client commands listed under Related Information. Each of these commands executes in the environment of one system partition. If the SP system is not partitioned, these commands execute in the environment of the entire system. In any case, the Syspar_map object class is used to determine which nodes are contained in the current environment. The attributes of interest are *syspar_name* and *node_number*.

Starting and Stopping the hardmon Daemon

The **hardmon** daemon is under System Resource Controller (SRC) control. It uses the signal method of communication in SRC. The **hardmon** daemon is a single subsystem and not associated with any SRC group. The subsystem name is **hardmon**. To start the **hardmon** daemon, use the **startsrc −s hardmon** command. This starts the daemon with the default arguments and SRC options. The **hardmon** daemon is setup to be respawnable and be the only instance of the **hardmon** daemon running on a control workstation. Do **not** start the **hardmon** daemon from the command line without using the **startsrc** command to start it.

To stop the **hardmon** daemon, use the **stopsrc −s hardmon** command. This stops the daemon and does not allow it to respawn.

To display the status of the **hardmon** daemon, use the **lssrc −s hardmon** command.

If the default startup arguments need to be changed, use the **chssys** command to change the startup arguments or the SRC options. Refer to *AIX Version 4 Commands Reference* and *AIX Version 4 General Programming Concepts: Writing and Debugging Programs* for more information about daemons under SRC control and how to modify daemon arguments when under SRC.

To view the current SRC options and daemon arguments, use the **odmget −q 'subsysname=hardmon' SRCsubsys** command.

Files

/spdata/sys1/spmon/hmthresholds
Contains boundary values.

/spdata/sys1/spmon/hmacls
Contains Access Control Lists.

| **/spdata/sys1/spmon/hmdceacls**
| DCE ACL database.

Location

/usr/lpp/ssp/bin/hardmon

Related Information

Commands: **hmadm**, **hmcmds**, **hmmon**, **nodecond**, **spmon**, **s1term**

File: **/spdata/sys1/spmon/hmacls**

Examples

1. To start the **hardmon** daemon, enter:
```
startsrc -s hardmon
```

2. To stop the **hardmon** daemon, enter:

```
stopsrc -s hardmon
```

3. To display the status of the **hardmon** daemon, enter:

```
lssrc -s hardmon
```

4. To display the status of all the daemons under SRC control, enter:

```
lssrc -a
```

5. To display the current SRC options and daemon arguments for the **hardmon** daemon, enter:

```
odmget -q 'subsysname=hardmon' SRCsubsys
```

hats Script

Purpose

hats – Starts or restarts Topology Services on a node or on the control workstation.

Syntax

hats

Flags

None.

Operands

None.

Description

Use this command to start the operation of Topology Services for a system partition (the **hatsd** daemon) on the control workstation or on a node within a system partition.

The **hats** script is not normally executed from the command line. It is normally called by the **hatsctrl** command, which is in turn called by the **syspar_ctrl** command during installation of the system, and partitioning or repartitioning of the system.

The Topology Services subsystem provides internal services to PSSP components.

Note that the **hats** script issues the **no -o nonlocsrcroute=1** command, which enables IP source routing. Do **not** change this setting, because the Topology Services subsystem requires this setting to work properly. If you change the setting, the Topology Services subsystem and a number of other subsystems that depend on it will no longer operate properly.

The **hatsd** daemon is initially started on the control workstation with the System Resource Controller (SRC), regardless of the level of the system partition. It is respawned automatically if the **hatsd** daemon encounters errors. The SP_NAME environment variable causes selection of the correct topology configuration.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

The “Starting Up and Shutting Down the SP System” chapter and “The System Data Repository” appendix in *PSSP: Administration Guide*

AIX Version 4 Commands Reference

Information about the System Resource Controller (SRC) in *AIX Version 4 General Programming Concepts: Writing and Debugging Programs*

Location

/usr/sbin/rsct/bin/hats

Related Information

Commands: **hatsctrl**, **lssrc**, **startsrc**, **stopsrc**, **syspar_ctrl**

Examples

See the **hatsctrl** command.

hatsctrl Script

Purpose

hatsctrl – A control script that starts the Topology Services subsystem.

Syntax

hatsctrl {–**a** | –**s** | –**k** | –**d** | –**c** | –**u** | –**t** | –**o** | –**r** | –**h**}

Flags

–**a**	Adds the subsystem.
–**s**	Starts the subsystem.
–**k**	Stops the subsystem.
–**d**	Deletes the subsystem.
–**c**	Cleans the subsystems, that is, delete them from all system partitions.
–**u**	Unconfigures the subsystems from all system partitions.
–**t**	Turns tracing on for the subsystem.
–**o**	Turns tracing off for the subsystem.
–**r**	Refreshes the subsystem.
–**h**	Displays usage information.

Operands

None.

Description

Topology Services is a distributed subsystem of PSSP that provides information to other PSSP subsystems about the state of the nodes and adapters on the IBM RS/6000 SP.

The **hatsctrl** control script controls the operation of the Topology Services subsystem. The subsystem is under the control of the System Resource Controller (SRC) and belongs to a subsystem group called **hats**. Associated with each subsystem is a daemon and a script that configures and starts the daemon.

An instance of the Topology Services subsystem executes on the control workstation and on every node of a system partition. Because Topology Services provides its services within the scope of a system partition, its subsystem is said to be system partition-sensitive. This control script operates in a manner similar to the control scripts of other system partition-sensitive subsystems. It can be issued from either the control workstation or any of the system partition's nodes.

From an operational point of view, the Topology Services subsystem group is organized as follows:

Subsystem	Topology Services
Subsystem Group	**hats**

SRC Subsystem **hats**

The **hats** subsystem is associated with the **hatsd** daemon and the **hats** script. The **hats** script configures and starts the **hatsd** daemon.

The subsystem name on the nodes is **hats**. There is one of each subsystem per node and it is associated with the system partition to which the node belongs.

On the control workstation, there are multiple instances of each subsystem, one for each system partition. Accordingly, the subsystem names on the control workstation have the system partition name appended to them. For example, for system partitions named **sp_prod** and **sp_test**, the subsystems on the control workstation are named **hats.sp_prod** and **hats.sp_test**.

Daemons **hatsd**

The **hatsd** daemon provides the Topology Services. The **hats** script configures and starts the **hatsd** daemon.

The **hatsctrl** script is not normally executed from the command line. It is normally called by the **syspar_ctrl** command during installation of the system, and partitioning or repartitioning of the system.

The **hatsctrl** script provides a variety of controls for operating the Topology Services subsystem:

- Adding, starting, stopping, and deleting the subsystem
- Cleaning up the subsystems, that is, deleting them from all system partitions
- Unconfiguring the subsystems from all system partitions
- Turning tracing on and off
- Refreshing the subsystem

Before performing any of these functions, the script obtains the current system partition name and IP address (using the **spget_syspar** command) and the node number (using the **node_number**) command. If the node number is zero, the control script is running on the control workstation.

Except for the clean and unconfigure functions, all functions are performed within the scope of the current system partition.

Adding the Subsystem

When the –**a** flag is specified, the control script uses the **mkssys** command to add the Topology Services subsystem to the SRC. The control script operates as follows:

1. It makes sure that the **hats** subsystem is stopped.
2. It gets the port number for the **hats** subsystem for this system partition from the **Syspar_ports** class of the System Data Repository (SDR) and ensures that the port number is set in the **/etc/services** file. If there is no port number in the SDR and this script is running on the control workstation, the script obtains a port number. If the script is running on a node and there is no port number in

the SDR, the script ends with an error. The range of valid port numbers is 10000 to 10100, inclusive.

The service name that is entered in the **/etc/services** file is **hats.***syspar_name*.

3. It checks to see if the subsystem is already configured in the SDR. If not, it creates an instance of the **TS_Config** class for this subsystem with default values. The default values are:
   - Heartbeats are sent out a rate of 1 per second (**Frequency** attribute = 1)
   - The number of heartbeats from the neighboring node that can be missed before the neighbor is declared inoperative is 4 (**Sensitivity** attribute = 4)
   - The execution priority is fixed (**Run_FixPri** attribute = 1)
   - The value of the execution priority used on the **set_priority** system call is 38 (**FixPri_Value** attribute = 38).
4. It removes the **hats** subsystem from the SRC (just in case it is still there).
5. It adds the **hats** subsystem to the SRC. On the control workstation, the IP address of the system partition is specified to be supplied as an argument to the daemon by the **mkssys** command.
6. It adds an entry for the **hats** group to the **/etc/inittab** file. The entry ensures that the group is started during boot. However, if **hatsctrl** is running on a High Availability Control Workstation (HACWS), no entry is made in the **/etc/inittab** file. Instead, HACWS manages starting and stopping the group.

Starting the Subsystem

When the **-s** flag is specified, the control script uses the **startsrc** command to start the Topology Services subsystem, **hats**.

Stopping the Subsystem

When the **-k** flag is specified, the control script uses the **stopsrc** command to stop the Topology Services subsystem, **hats**.

Deleting the Subsystem

When the **-d** flag is specified, the control script uses the **rmssys** command to remove the Topology Services subsystem from the SRC. The control script operates as follows:

1. It makes sure that the **hats** subsystem is stopped.
2. It removes the **hats** subsystem from the SRC using the **rmssys** command.
3. It removes the port number from the **/etc/services** file.
4. If there are no other subsystems remaining in the **hats** group, it removes the entry for the **hats** group from the **/etc/inittab** file.

Cleaning Up the Subsystems

When the **-c** flag is specified, the control script stops and removes the Topology Services subsystems for all system partitions from the SRC. The control script operates as follows:

1. It stops all instances of subsystems in the subsystem group in all partitions, using the **stopsrc -g hats** command.
2. It removes the entry for the **hats** group from the **/etc/inittab** file.
3. It removes all instances of subsystems in the subsystem group in all partitions from the SRC using the **rmssys** command.
4. It removes all entries for the **hats** subsystems from the **/etc/services** file.

Unconfiguring the Subsystems

When the −**u** flag is specified, the control script performs the function of the −**c** flag in all system partitions and then removes all port numbers from the SDR allocated by the Topology Services subsystems.

Note: The −**u** flag is effective only on the control workstation.

Prior to executing the **hatsctrl** command with the −**u** flag on the control workstation, the **hatsctrl** command with the −**c** flag must be executed from all of the nodes. If this subsystem is not successfully cleaned from all of the nodes, different port numbers may be used by this subsystem, leading to undefined behavior.

Turning Tracing On

When the −**t** flag is specified, the control script turns tracing on for the **hatsd** daemon, using the **traceson** command.

Turning Tracing Off

When the −**o** flag is specified, the control script turns tracing off (returns it to its default level) for the **hatsd** daemon, using the **tracesoff** command.

Refreshing the Subsystem

When the −**r** flag is specified, the control script refreshes the subsystem, using the **hats refresh** command and the **refresh** command. It rebuilds the information about the node and adapter configuration in the SDR and signals the daemon to read the rebuilt information. To refresh the subsystem across all nodes execute **hatsctrl** −**r** from the control workstation.

Logging

While it is running, the Topology Services daemon provides information about its operation and errors by writing entries in a log file. The **hatsd** daemon in the system partition named *syspar_name* uses a log file called **/var/ha/log/hats.***syspar_name*.

Files

/var/ha/log/hats.*syspar_name***.**
Contains the log of the **hatsd** daemon on the system partition named *syspar_name*.

Standard Error

This command writes error messages (as necessary) to standard error.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error occurred.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

AIX Version 4 Commands Reference

Information about the System Resource Controller (SRC) in *AIX Version 4 General Programming Concepts: Writing and Debugging Programs*

Location

/usr/sbin/rsct/bin/hatsctrl

Related Information

Commands: **hats**, **lssrc**, **startsrc**, **stopsrc**, **syspar_ctrl**

Examples

1. To add the Topology Services subsystem to the SRC in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   hatsctrl -a
   ```

2. To start the Topology Services subsystem in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   hatsctrl -s
   ```

3. To stop the Topology Services subsystem in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   hatsctrl -k
   ```

4. To delete the Topology Services subsystem from the SRC in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   hatsctrl -d
   ```

5. To clean up the Topology Services subsystem on all system partitions, enter:

   ```
   hatsctrl -c
   ```

6. To unconfigure the Topology Services subsystem from all system partitions, on the control workstation, enter:

   ```
   hatsctrl -u
   ```

7. To turn tracing on for the Topology Services daemon in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   hatsctrl -t
   ```

8. To turn tracing off for the Topology Services daemon in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   hatsctrl -o
   ```

9. To display the status of all of the subsystems in the Topology Services SRC group, enter:

   ```
   lssrc -g hats
   ```

10. To display the status of an individual Topology Services subsystem, enter:

    lssrc -s *subsystem_name*

11. To display detailed status about an individual Topology Services subsystem, enter:

    lssrc -l -s *subsystem_name*

    In response, the system returns information that includes the running status of the subsystem, the number of defined and active nodes, the required number of active nodes for a quorum, the status of the group of nodes, and the IP addresses of the source node, the group leader, and the control workstation.

12. To display the status of all of the daemons under SRC control, enter:

    ```
    lssrc -a
    ```

hatsoptions

Purpose

hatsoptions – Controls Topology Services options on a node or control workstation.

Syntax

hatsoptions [**-s**] [**-d**]

Flags

–s Instructs the Topology Services daemon to reject messages that are apparently delayed.

–d Instructs the Topology Services daemon not to reject messages that are apparently delayed (this is the default).

Operands

None.

Description

Before this command can be executed, environment variable HB_SERVER_SOCKET must be set to the location of the UNIX-domain socket used by the Topology Services subsystem. The statement below can be used:

```
export HB_SERVER_SOCKET=/var/ha/soc/hats/server_socket.partition name
```

Alternatively, variable HA_SYSPAR_NAME can be set to the partition name.

The Topology Services daemon must be running in order for this command to be successful.

hatsoptions can be used to control a number of options in Topology Services. Option **-s** instructs the Topology Services daemon to reject messages that are apparently delayed. This can be used in very large system configurations, where messages are sometimes delayed in the network or in the sender and receiver nodes. Use this option only if the Time-Of-Day clocks are synchronized across all the nodes and the control workstation. Otherwise messages may be incorrectly discarded when the sender's Time-Of-Day clock is behind the receiver's.

Option **-d** instructs the Topology Services daemon not to reject messages that are apparently delayed. This is the default.

Environment Variables

HB_SERVER_SOCKET
This environment variable should be set before this command can be executed. It must be set to the location of the UNIX-domain socket used by Topology Services clients to connect to the Topology Services daemon. This environment variable must be set to **/var/ha/soc/hats/server_socket./var/ha/soc/hats/server_socket.***partition name*.

HA_SYSPAR_NAME
If HB_SERVER_SOCKET is not set, then HA_SYSPAR_NAME must be set to the partition name.

Files

/var/ha/soc/hats/server_socket.*partition name*

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Exit Values

0 Indicates the successful completion of the command.

1 Indicates the command was unsuccessful.

Security

You must have root privilege to run this command.

Prerequisite Information

AIX Version 4 Commands Reference

Location

/usr/sbin/rsct/bin/hatsoptions

Related Information

Commands: **hatsctrl**, **hats**, **lssrc**, **startsrc**, **stopsrc**, **syspar_ctrl**

Examples

To instruct the Topology Services daemon on the local node to start discarding apparently delayed messages, enter:

```
export HA_SYSPAR_NAME=partition1

     /usr/sbin/rsct/bin/hatsoptions -s
```

hatstune

Purpose

hatstune – Tunes HATS (High Availability Topology Services) parameters.

Syntax

hatstune [**−f** *frequency*] [**−s** *sensitivity*] [**−p** *priority*]
[**−l** *log_length*] [**−m** *pin_object*] [**−r**]
hatstune **−d** [**−r**]
hatstune **−v**
hatstune **−h**

Flags

−f *frequency*
: Set HATS heart beat frequency (interval between heart beats). The frequency is an integer value of heart beat frequency in seconds. The valid frequency range is [1, 30] or a special keyword "default." The default frequency value is used if the frequency is "default."

−s *sensitivity*
: Set HATS heart beat sensitivity; the maximum number of missing heart beats allowed before the adapter is declared down.

The *sensitivity* is an integer value of heart beat sensitivity. The valid sensitivity range is [4, 40] or a special keyword "default." The default sensitivity value will be used if sensitivity is the special keyword "default."

−p *priority*
: Specifies the AIX process scheduling priority at which the HATS daemon should be run. Valid priority values are: the [10, 80] range, the special value 0, and the "default" keyword. The HATS daemon will run on the HATS default fixed priority if the special keyword "default" is specified. The HATS daemon will run on the AIX default user program non-fixed priority if the special priority value 0 is used.

−l *log_length*
: Sets the maximum user and service log file length. The *log_length* is the maximum number of lines in the log files. The valid range is [2000, 1000000] or a special keyword "default." The default log length value will be used if *log_length* is "default."

−m *pin_object*
: Specifies whether the Topology Services daemon should be pinned in memory. Because HATS is a real time program, it should be able to run on a timely basis. Pinning some of HATS' address space in memory reduces the likelihood that the Topology Services daemon could become blocked for paging operations.

This option specifies which HATS objects should be pinned in real memory. The valid objects are data, text, proc, none, and a special keyword "default."

data Only the data segment will be pinned in real memory.

text Only the code segment will be pinned in real memory.

proc Both data and text will be pinned in real memory.

none None of above will be pinned in real memory.

default
The default setting will be used.

The above *pin_objects* are mutually exclusive. The objects are not case sensitive.

−r Refreshes HATS after the tunable parameters are set successfully. By default, hatstune sets the tunable parameters only. The new parameters do not take effect until a HATS refresh is performed.

−d Resets all HATS tunable parameters to their default values.

−v Displays current settings of all HATS tunable parameters.

−h Prints a brief usage message.

Operands

None.

Description

HATS tunable parameters can be set by changing SDR attributes of certain SDR objects. This command offers an easier mechanism to change HATS tunable parameters. Besides not requiring knowledge of the SDR objects needed to tune HATS, **hatstune** performs consistency checking on the tunable values.

The valid and default values of HATS tunable parameters are:

Heart beat frequency: (seconds)
1 <= Frequency <= 30 (default: 1)

Heart beat sensitivity:
4 <= Sensitivity <= 40 (default: 4)

Run HATS daemon at a fixed priority:
10 <= FixPri_Value <= 80 (default: 38)

Maximum log file length: (lines)
2000 <= Log_Length <= 1000000 (default: 5000)

Objects to be pinned in real memory:
Data, Text, Proc (Data and Text), None (default: Text)

Multiple options can be used together. The same option cannot be specified more than once.

By default, the new HATS tunable values are not in effect until a refresh operation is done by using the −**r** option or until a **hatsctrl** −**r** command is issued. This allows you to change the HATS tunable parameters as many times as desired and then specify the refresh option to have the changes take effect.

Value checking (−**v** option) can be performed on any SP node by any user. Value assignment (−**f**,−**s**, −**p**, −**l**, −**m**, −**d** options) can only be run on the control workstation by users who have root privilege.

Environment Variables

The **SP_NAME** variable can be used to designate the applicable partition.

Standard Output

Usage message, current HATS tunable values, new tunable values.

Standard Error

Output consists of error messages, when the command cannot complete successfully.

Exit Values

0 Indicates successful completion of the command.

nonzero Indicates that an error occurred.

Security

The command can be run by any user on any SP node to check the current HATS tunable values. However, only users who have root privilege can change the settings.

Restrictions

Value checking (−**v** option) can be performed on any SP node by any user. Value assignment (−**f**,−**s**, −**p**, −**l**, −**m**, −**d** options) can only be run on the control workstation by users who have root privilege.

Implementation Specifics

This command is part of the IBM Reliable Scalable Cluster Technology (RSCT) on the RS/6000 system.

Prerequisite Information

"The Topology Services Subsystem" chapter in the *PSSP: Administration Guide*.

Location

/usr/sbin/rsct/bin/hatstune

Related Information

Commands: **hatsctrl**, **SDRGetObjects**, **SDRChangeAttrValues**, **splst_syspars**

Examples

1. To change the heart beat frequency to 3 seconds and the heart beat sensitivity to 10 (this command must be run on the control workstation), enter:

   ```
   hatstune -f 3 -s 10
   ```

2. To change the maximum log file lengths to 10000 lines and cause the change to take effect immediately (this command must be run on the control workstation), enter:

   ```
   hatstune -l 10000 -r
   ```

3. To reset all HATS tunable parameters to default values (this command must be run on the control workstation), enter:

```
hatstune -d
```

4. To display current HATS tunable settings, enter:

```
hatstune -v
```

5. To show the usage message, enter:

```
hatstune -h
```

hc.vsd

Purpose

hc.vsd – Queries and controls the hc daemon of the Recoverable Virtual Shared Disk subsystem.

Syntax

hc.vsd {**CLIENT_PATH** *socket_path* | **debug** [**off**] | **mksrc** | **PING_DELAY** *delay_in_sec* | **query** | **qsrc** | **reset** | **rmsrc** | **SCRIPT_PATH** *de/activate_path* | **start** | **stop** | **trace [off]**}

Flags

None.

Operands

CLIENT_PATH *socket_path*
: Specifies the path for the socket connection to the hc client. The default is **/tmp/serv**.

debug [off]
: Specify debug to redirect the hc subsystem's stdout and stderr to the console and cause the hc subsystem to not respawn if it exits with an error. (You can use the **lscons** command to determine the current console.)

The hc subsystem must be restarted for this operand to take effect

Once debugging is turned on and the hc subsystem has been restarted, **hc.vsd trace** should be issued to turn on tracing

Use this operand under the direction of your IBM service representative.

Note: the default when the node is booted is to have stdout and stderr routed to the console. If debugging is turned off stdout and stderr will be routed to **/dev/null** and all further trace messages will be lost. You can determine if debug has been turned on by issuing **hc.vsd qsrc**. If debug has been turned on the return value will be:

```
action = "2"
```

mksrc
: Uses **mkssys** to create the hc subsystem.

PING_DELAY *delay_in_sec*
: Specifies the time in seconds between pings to the hc client. The default is 600 seconds.

query
: Displays the current status of the hc subsystem in detail.

qsrc
: Displays the System Resource Controller (SRC) configuration of the HC daemon.

reset
: Stops and restarts the hc subsystem.

rmsrc
: Uses **rmssys** to remove the hc subsystem.

SCRIPT_PATH *de/activate_path*
: Specifies the location of the user-supplied scripts to be run when hc activates or deactivates.

start Starts the hc subsystem.

stop Stops the hc subsystem.

trace [off] Requests or stops tracing of the hc subsystem. The hc subsystem must be in the active state when this command is issued.

This operand is only meaningful after the **debug** operand has been used to send stdout and stderr to the console and the hc subsystem has been restarted.

Description

Use this command to display information about the hc subsystem and to change the status of the subsystem.

You can restart the hc subsystem with the IBM Virtual Shared Disk Perspective. Type `spvsd` and select actions for virtual shared disk nodes.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error occurred.

Note: The **query** and **qsrc** subcommands have no exit values.

Security

You must have root privilege to issue the **debug**, **mksrc**, **reset**, **start**, and **stop** subcommands.

Implementation Specifics

This command is part of the IBM Recoverable Virtual Shared Disk option of PSSP.

Prerequisite Information

See *PSSP: Managing Shared Disks*

Location

/usr/lpp/csd/bin/hc.vsd

Related Information

Commands: **ha_vsd**, **ha.vsd**

Examples

To stop the hc subsystem and restart it, enter:

```
hc.vsd reset
```

The system returns the messages:

```
Waiting for the hc subsystem to exit.

hc subsystem exited successfully.

Starting hc subsystem.

hc subsystem started PID=xxx.
```

hmadm

Purpose

hmadm – Administers the Hardware Monitor daemon.

Syntax

hmadm [{–**d** *debug_flag*} ...] *operation*

Flags

–**d** *debug_flag*
Specifies the daemon debug flag to be set or unset in the daemon.

Operands

operation Specifies the administrative action to perform.

The *operation* must be one of the following:

cleard Unsets the daemon debug flag specified by the –**d** flag in the daemon. Multiple –**d** flags can be specified. If no –**d** flags are specified, the **all** debug flag is assumed.

clog Changes the daemon log file. If the log file is growing large, this operation is used to cause the daemon to write to a new log file.

quit Causes the daemon to exit.

setacls This option applies only to Kerberos 4 Compatibility authentication method. Reads the Hardware Monitor access control list configuration file to update the daemon's internal ACL tables. Any Hardware Monitor application or command executing under the ID of a user who has changed or deleted ACLs has its client connection terminated by the daemon. Such applications and commands must be restarted, if possible. ACLs for new users can be added without any effect on executing applications and commands.

This operation must by invoked by the administrator after the administrator modifies the ACL configuration file.

setd Sets the daemon debug flag specified by the –**d** flag in the daemon. Multiple –**d** flags can be specified. If no –**d** flags are specified, the **all** debug flag is assumed.

Description

The **hmadm** command is used to administer the Hardware Monitor daemon. The Hardware Monitor daemon executes on the control workstation and is used to monitor and control the SP hardware. Five administrative actions are supported, as specified by the *operation* operand.

Normally when the daemon exits, it is automatically restarted by the system. If frame configuration information is changed, the **quit** operation can be used to update the system.

The daemon writes debug information and certain error information to its log file. The log file is located in **/var/adm/SPlogs/spmon** and its name is of the form **hmlogfile.***nnn*, where *nnn* is the Julian date of the day the log file was opened by the daemon. The **clog** operation causes the daemon to close its current log file and create a new one using the name **hmlogfile***nnn*, where *nnn* is the current Julian date. If this name already exists, a name of the form **hmlogfile.***nnn_m* is used, where *m* is a number picked to create a unique file name.

The s70 daemon writes debug information and certain error information to its log file. The log file is located in **/var/adm/SPlogs/spmon/s70d** and is of the form **s70d.***f***.log.***ddd*, where *f* is the frame number, and *ddd* is the Julian date of the day the log was opened by the s70 daemon.

The **hmadm clog** operation causes the s70 daemon, as well as **hardmon**, to close its current log file and open a new one. For the s70 daemon the form of the new log file name is the same as the original log file unless that name already exists, in which case the form becomes **s70d.***f***.log.***ddd_n*, where *n* is an incremental number picked to create a unique file name.

The Netfinity daemon writes debug information and certain error information to its log file. The log file is located in **/var/adm/SPlogs/spmon/nfd** and is of the form **nfd.***f***.log.***ddd*, where *f* is the frame number, and *ddd* is the Julian date of the day the log was opened by the Netfinity daemon.

The **hmadm clog** operation causes the Netfinity daemon, as well as **hardmon**, to close its current log file and open a new one. For the Netfinity daemon the form of the new log file name is the same as the original log file unless that name already exists, in which case the form becomes **nfd.***f***.log.***ddd_n*, where *n* is an incremental number picked to create a unique file name.

The following debug flags are supported by the daemon:

Table 5.

debug_flag	**hardmon**	**s70**	**nfd**
all	Set/unsets all of hardmon's debug options	Sets/unsets all of s70d's debug options	Sets/unsets all of nfd's debug options
acls	Logs the Access Control Lists	N/A	N/A
cmdq	Logs the contents of the internal queue of commands sent to the frames	N/A	N/A
cntrs	Logs the daemon (hardmon) internal counters	Logs the daemon (s70d) internal counters	Logs the daemon (nfd) internal counters
dcmds	Logs commands sent to the daemon (hardmon)	Logs commands sent to the daemon (s70d)	Logs commands sent to the daemon (nfd)
fcmds	Logs commands sent to the frame supervisors	Logs commands sent to the S70 hardware	Logs commands sent to the Netfinity hardware
ipl	Logs interested party lists	N/A	N/A
memchk	Logs memory allocation data in **hm_memory_dump**	N/A	N/A
pckts	Logs packets received from the frame supervisors in **hm_frame_packet_dump**	Logs packets sent to hardmon in **s70d.*f*.packet.dump** where *f* is the frame number	Logs packets sent to hardmon in **nfd.*f*.packet.dump** where *f* is the frame number
polla	Logs poll list array	N/A	N/A
rsps	Logs responses sent to clients in **hm_response_dump**	N/A	N/A
sami	N/A	Logs data sent to and received from the S70 Control Panel in **s70d.*f*.sami_dump** where *f* is the frame number	Logs data sent to and received from the Netfinity Control Panel in **nfd.*f*.slim_dump** where *f* is the frame number
socb	Logs client socket session information.	N/A	N/A
s1data	Logs data sent to the S1 serial ports in **hm_s1data_dump**	Logs data sent to and received from the S70 serial port in **s70d.*f*.s1data_dump** where *f* is the frame number	Logs data sent to and received from the Netfinity serial port in **nfd.*f*.s1data_dump** where *f* is the frame number
s1refs	Logs S1 serial port reference counts and connections	N/A	N/A
ttycb	Logs ttycb control blocks	N/A	N/A
tvars	Logs boundary values used in checking temperatures, amperages, and volts	N/A	N/A

Security

This command uses the SP Hardware Monitor. To execute this command, users must be authorized to access the Hardware Monitor subsystem with administrative permission.

Location

/usr/lpp/ssp/bin/hmadm

Related Information

Commands: **hmdceobj**, **hmckacls**, **hmgetacls**

Files: **/spdata/sys1/spmon/hmdceacls**, **/spdata/sys1/spmon/hmacls**

Examples

1. To write hardmon client socket session debug information to hardmon's log file, enter:

   ```
   hmadm -d socb setd
   ```

2. To write debug information that hardmon and s70 daemon send to the frames in their respective log files, enter:

   ```
   hmadm -d fcmds setd
   ```

   To clear this debug flag, enter:

   ```
   hmadm -d fcmds cleard
   ```

3. To clear all debug flags previously set, enter:

   ```
   hmadm cleard
   ```

4. To change hardmon and s70 daemon log files, enter:

   ```
   hmadm clog
   ```

5. To cause hardmon to exit, which also causes the s70 daemon to exit, enter:

   ```
   hmadm quit
   ```

hmckacls

Purpose

hmckacls – Checks the existence of Hardware Monitor permissions.

Syntax

hmckacls [–**f** *file_name*] *permission_string* [*slot_spec*]

Flags

–**f** *file_name*
: Uses the *file_name* as the source of the slot ID specifications.

Operands

permission_string
: The *permission_string* is one or more characters taken from the set **v**, **s**, **m**, and **u**. A definition of each follows:

  v Specifies Virtual Front Operator Panel (VFOP) permission.

  s Specifies S1 (serial) permission.

  m Specifies Monitor permission. The VFOP permission implies Monitor permission.

  u Specifies Microcode Download permission. (**u** is only valid when DCE is a configured authentication method).

slot_spec Specifies the addresses of the hardware components.

Description

The **hmckacls** command checks the existence of Hardware Monitor permissions that are associated with the specified hardware. The command returns **0** if the ACL(s) in the specified permission string exist for all of the slots contained in the target slot specification. It returns **1** if any of the ACLs do not exist in any one of the targets in the slot specification.

Exit Values

0 Indicates the ACL(s) in the specified permission string exist.

1 Indicates one or more ACLs in the specified permission string does not exist.

Location

/usr/lpp/ssp/bin/hmckacls

Related Information

Commands: **lsauthts**, **chauthts**, **hmgetacls**, **hmdceobj**

Files: **/spdata/sys1/spmon/hmacls**, **/spdata/sys1/spmon/hmdceacls**

Examples

In the following examples the same command **(hmckacls 1:1**) is issued four times; once for each of the possible authorization method configurations. The commands target is slot 1 in frame 1.

1. If the configured authentication method is DCE only then the ACLs are applied at the slot level. In this example, slot 1 has **v**, **s**, **m**, and **u** permissions. The return reflects that all of the permissions exist for frame 1, slot 1:

   ```
   [/]> hmckacls vsmu 1:1

   [/]> echo $?

   0
   ```

2. If the configured authentication method is COMPAT only then the ACLs are applied at the frame level. Therefore, all slots within a frame will have the same permissions as does the frame. In this example the frame has **v**, **s**, and **m** permissions (**u** is only valid in DCE mode). The return shows that the **u** permission does not exist for frame 1, slot 1:

   ```
   [/]> hmckacls vsmu 1:1

   [/]> echo $?

   1
   ```

3. If the configured authentication method is both DCE and COMPAT then the ACLs that are applied is the result of performing an "or" operation on both DCE and COMPAT permissions. The return reflects that all of the permissions exist for frame 1, slot 1:

   ```
   [/]> hmckacls vsmu 1:1

   [/]> echo $?

   0
   ```

4. If the configured authentication method is neither DCE nor COMPAT (there is no configured method), then access is granted based on whether the user is root and on the control workstation.

   If the user is root and on the control workstation, the return shows that the user can perform all of the actions:

   ```
   [/]> hmckacls vsmu 1:1

   [/]> echo $?

   0
   ```

   If the user is not root or not on the control workstation, the return shows that at least one of the permissions is not valid:

   ```
   [/]> hmckacls vsmu 1:1

   [/]> echo $?

   1
   ```

hmcmds

Purpose

hmcmds – Controls the state of the SP hardware.

Syntax

hmcmds [−**a** | −**v**] [−**f** *file_name*] [−**u** *microcode_file_name*]
[−**G**] *command* [*slot_spec* ... | **all**]

Flags

−**a** Exits immediately after sending the **VFOP** command to the specified hardware; that is, it does not wait for the hardware state to match the command.

−**v** Specifies verbose mode. The percentage of hardware components whose state matches the **VFOP** command is displayed at five-second intervals. The following are also displayed:

- The number of components to which the **VFOP** command was sent
- A list of components which did not change state
- The number of components expected to change state
- The number of components which achieved the expected state
- The number of components to which the **VFOP** command was not sent
- A list of components to which the **VFOP** command was not sent

−**f** *file_name*
Uses *file_name* as the source of slot ID specifications.

−**u** *microcode_file_name*
Uses *microcode_file_name* as the source of supervisor microcode that is loaded to the specified *slot_spec*. If the *microcode_file_name* is not fully qualified, the file must be in the current directory. This option is allowed only with the **microcode** command.

−**G** Specifies Global mode. With this flag, commands can be sent to any hardware.

Operands

command Specifies the command to send to the hardware components.

slot_spec Specifies the addresses of the hardware components.

Description

Use this command to control the state of the SP hardware. Control is provided via the Virtual Front Operator Panel (VFOP). VFOP is a set of commands that can be sent to the hardware components contained in one or more SP frames. Each frame consists of 18 slots, numbered 0 through 17, where slot 0 represents the frame itself, slot 17 can contain a switch and slots 1 through 16 can contain thin or wide processing nodes. Wide nodes occupy two slots and are addressed by the odd slot number. In a switch only frame, slots 1 through 16 can contain switches; the switches occupy two slots and are addressed by the even slot number.

Normally, commands are only sent to the hardware components in the current system partition. A system partition only contains processing nodes. The switches and the frames themselves are not contained in any system partition. To send **VFOP** commands to hardware components not in the current system partition or to any frame or switch, use the –**G** flag.

The following list describes the **VFOP** command set. Commands that require the –**G** flag are marked by an asterisk (*). Commands marked by a double asterisk (**) are primarily used by the **Eclock** command and are not intended for general use since an in-depth knowledge of switch clock topology is required to execute these commands in the proper sequence.

Before issuing these commands, refer to the “Using a Switch” chapter in the *PSSP: Administration Guide* for detailed descriptions.

SP Switch

clkdrv2 Sets the SP Switch clock drive to the Phase Lock Loop 2.**

clkdrv3 Sets the SP Switch clock drive to the Phase Lock Loop 3.**

clkdrv4 Sets the SP Switch clock drive to the Phase Lock Loop 4.**

clkdrv5 Sets the SP Switch clock drive to the Phase Lock Loop 5.**

hold_power_reset
Performs power-on reset of SP Switch and holds the SP Switch in reset state. Requires **rel_power_reset** to release.**

hold_synch_reset
Performs synchronous reset of SP Switch and holds the SP Switch in reset state. Requires **rel_synch_reset** to release.**

intclk2 Sets the SP Switch clock input to the Local Oscillator 2.**

intclk4 Sets the SP Switch clock input to the Local Oscillator 4.**

jack3 Sets the SP Switch clock input to the External Jack 3.**

jack4 Sets the SP Switch clock input to the External Jack 4.**

jack5 Sets the SP Switch clock input to the External Jack 5.**

jack6 Sets the SP Switch clock input to the External Jack 6.**

jack7 Sets the SP Switch clock input to the External Jack 7.**

jack8 Sets the SP Switch clock input to the External Jack 8.**

jack9 Sets the SP Switch clock input to the External Jack 9.**

jack10 Sets the SP Switch clock input to the External Jack 1.**

jack11 Sets the SP Switch clock input to the External Jack 11.**

jack12 Sets the SP Switch clock input to the External Jack 12.**

jack13 Sets the SP Switch clock input to the External Jack 13.**

jack14 Sets the SP Switch clock input to the External Jack 14.**

jack15 Sets the SP Switch clock input to the External Jack 15.**

jack16 Sets the SP Switch clock input to the External Jack 16.**

jack17 Sets the SP Switch clock input to the External Jack 17.**

jack18 Sets the SP Switch clock input to the External Jack 18.**

jack19 Sets the SP Switch clock input to the External Jack 19.**

jack20 Sets the SP Switch clock input to the External Jack 20.**

jack21 Sets the SP Switch clock input to the External Jack 21.**

jack22 Sets the SP Switch clock input to the External Jack 22.**

jack23 Sets the SP Switch clock input to the External Jack 23.**

jack24 Sets the SP Switch clock input to the External Jack 24.**

jack25 Sets the SP Switch clock input to the External Jack 25.**

jack26 Sets the SP Switch clock input to the External Jack 26.**

jack27 Sets the SP Switch clock input to the External Jack 27.**

jack28 Sets the SP Switch clock input to the External Jack 28.**

jack29 Sets the SP Switch clock input to the External Jack 29.**

jack30 Sets the SP Switch clock input to the External Jack 30.**

jack31 Sets the SP Switch clock input to the External Jack 31.**

jack32 Sets the SP Switch clock input to the External Jack 32.**

jack33 Sets the SP Switch clock input to the External Jack 33.**

jack34 Sets the SP Switch clock input to the External Jack 34.**

power_on_reset
Performs power-on reset of SP Switch. Includes chip self-test and synchronous reset.**

rel_power_reset
Releases SP Switch from **hold_power_reset** state.**

rel_synch_reset
Releases SP Switch from **hold_synch_reset** state.**

synch_reset
Performs synchronous reset of SP Switch. Turns off error enables and clears errors.**

Any Frame, Node, or Switch that Supports Microcode Download

basecode
Performs a switch power off of the node and switches the active frame, node, or switch supervisor to basecode mode causing the active supervisor to become nonactive and the basecode supervisor to become active.*

Note: You must issue this command before issuing the **microcode** command.

| **boot_supervisor [reboot (as alias)]**
Performs a boot of the frame, node, or switch basecode application and supervisor.*

stop_supervisor [stopsup (as alias)]
Causes an s70d daemon or nfd daemon to exit, without restarting.

exec_supervisor [startsup (as alias)]
Causes the basecode to execute the nonactive frame, node, or switch supervisor, making it active.*

microcode [ucode (as alias)]
Performs a download of supervisor microcode to the frame, node, or switch.*

Note: You must issue the **basecode** command before issuing this command.

Any Node, with the exception of Node Expansion Nodes such as S70, S80, and Netfinity, which do not support normal, secure, or service.

normal Sets the keylock on a processing node to the Normal position.

reset Presses and releases the reset button on a processing node.

secure Sets the keylock on a processing node to the Secure position.

service Sets the keylock on a processing node to the Service position.

Any Frame, with the exception of Node Expansion Nodes such as S70, S80, and Netfinity, which do not support setid.

runpost Initiates Power-On Self Tests (POST) in the frame supervisor.*

setid Sets the frame ID into the frame supervisor.*

Any Frame, Node, or Switch, with the exception of Node Expansion Frames such as S70, S80, and Netfinity, which do not support off, or on.

off Disables power to the frame power supplies, a processing node, or a switch.

on Enables power to the frame power supplies, a processing node, or a switch.

Any Node, or Switch, with the exception of Node Expansion Nodes such as S70, S80, and Netfinity, which do not support flash.

flash Flashes the I2C address of a processing node or a switch node in the node's yellow LED.

One of these commands must be specified using the *command* operand. The command is sent to the hardware specified by the *slot_spec* operands. However, the command is not sent to any hardware that is not in the current system partition unless the −**G** flag is specified. If the −**G** flag is not specified and the *slot_spec* operands specify no hardware in the current system partition, an error message is displayed.

The *slot_spec* operands are interpreted as slot ID specifications. A slot ID specification names one or more slots in one or more SP frames and it has either of two forms:

```
fidlist:sidlist   or   nodlist
```

where:

fidlist = fval[,fval,...]

sidlist = sval[,sval,...]

nodlist = nval[,nval,...]

The first form specifies frame numbers and slot numbers. The second form specifies node numbers. A *fval* is a frame number or a range of frame numbers of the form *a–b*. A *sval* is a slot number from the set 0 through 17 or a range of slot numbers of the form *a–b*. A *nval* is a node number or a range of node numbers of the form *a–b*.

The relationship of node numbers to frame and slot numbers is shown in the following formula:

node_number = ((*frame_number* – 1) × 16) + *slot_number*

Note: Node numbers can only be used to specify slots 1 through 16 of any frame.

The following are some examples of slot ID specifications.

To specify slot 1 in frames 1 through 10, enter:

```
1-10:1
```

To specify frames 2, 4, 5, 6, and 7, enter:

```
2,4-7:0
```

To specify slots 9 through 16 in frame 5, enter:

```
5:9-16
```

If frame 5 contained wide nodes, the even slot numbers are ignored.

To specify specifies slots 1, 12, 13, 14, 15, and 16 in each of frames 3 and 4, enter:

```
3,4:1,12-16
```

To specify slot 17 in frame 4, enter:

```
4:17
```

To specify the nodes in slots 1 through 16 of frame 2, enter:

```
17-32
```

To specify the nodes in slot 1 of frame 1, slot 1 of frame 2 and slot 1 of frame 3, enter:

```
1,17,33
```

To specify the node in slot 6 of frame 1, enter:

```
6
```

Optionally, slot ID specifications can be provided in a file rather than as command operands. The file must contain one specification per line. The command requires that slot ID specifications be provided. If the command is to be sent to all SP hardware, the keyword **all** must be provided in lieu of the *slot_spec* operands. However, the **all** keyword can only be specified if the –**G** flag is specified and if the

VFOP command is **on** or **off**, since **on** or **off** are the only commands common to all hardware components.

Commands sent to hardware for which they are not appropriate, or sent to hardware which does not exist, are silently ignored by the Hardware Monitor subsystem.

By default, and except for the **reset**, **flash**, and **run_post** commands, the **hmcmds** command does not terminate until the state of the hardware to which the command was sent matches the command or until 15 seconds have elapsed. If 15 seconds have elapsed, the **hmcmds** command terminates with a message stating the number of nodes whose state was expected to match the **VFOP** command sent and the number of nodes which actually are in that state. The state of hardware for which the **VFOP** command is inappropriate, or where the hardware does not exist, is ignored.

The **hmcmds** command fails if any of the hardware targeted is not supported by the version of PSSP on the machine where the command was issued.

Security

To execute the **hmcmds** command, the user must be authorized to access the Hardware Monitor subsystem and must be granted "VFOP" permission for the hardware objects (frames, slots) specified in the command. Commands sent to hardware objects for which the user does not have "VFOP" permission are ignored.

Location

/usr/lpp/ssp/bin/hmcmds

Related Information

Commands: **hmdceobj**, **hmmon**, **spsvrmgr**

Files: **/spdata/sys1/spmon/hmdceacls**, **/spdata/sys1/spmon/hmacls**

Examples

1. To turn power off in all hardware, enter:

   ```
   hmcmds -G off all
   ```

2. In a five-frame SP system, to set the keyswitch on all processing nodes to Secure, enter:

   ```
   hmcmds secure 1-5:1-16
   ```

3. To set the clock multiplexor in the switches in frames 1 through 8 to external clock 3, enter:

   ```
   hmcmds -G extclk3 1-8:17
   ```

4. In a three-frame SP system, to set the keyswitch to Normal on node 6 and on the nodes in slot 2 of both frames 2 and 3, enter:

   ```
   hmcmds normal 6 2,3:2
   ```

hmdceobj

Purpose

hmdceobj – Adds, deletes and lists hardware monitor DCE objects.

Syntax

hmdceobj {−**a** [−**g** *group_name* [−**p** *permission*]] | −**d** | −**q** | −**r** | −**s** [−**v**]}
{−**f** *file_name* | *hardware identifier...*}
hmdceobj −**h**

Flags

−a Specifies an add operation.

g *group_name*
Refers to a DCE group name that will be added to the default ACL. If not specified, the default ACL will be used. The default ACL is obtained from an object's parent (also called a "container" in DCE) in hardmon's ACL hierarchy. A slot's parent is the frame object, if it exists for that slot, or the system object. A frame's object is the system object. The system object is the highest in the hierarchy.

−p Specifies the permission list being set for the additional DCE group name that was specified with the −**g** flag. If not specified, the group will be added to the ACL without any permissions.

The permission list is a string of one or more of the following characters:

v VFOP (virtual front panel operator)

s s1 link permission

m monitor permission

u microcode update permission

a administrative permission

t check permission

−d Indicates that a delete operation is to be done.

−f *file_name*
Uses *file_name* as the source of hardware identifiers.

−q Lists all of the objects currently defined in the hardware monitor DCE database.

−r Outputs objects in raw form.

−s Outputs objects in symbolic form.

−v Displays verbose output.

−h Displays command syntax (help).

Operands

hardware identifier
Specifies the addresses of the hardware components. This may be a hardware monitor slot_spec or a node list.

Description

Use the **hmdceobj** command to add and delete hardmon DCE objects. The DCE objects represent hardware that the hardware monitor is responsible for on an SP system. Access Control Lists (ACLs) can be associated with the objects to control access. The **spacl** command or RS6000/SP Security SMIT interface is used to change the ACLs associated with an existing object. Hardware monitor protected objects that may be added or deleted are:

- Frames – containers for slot objects
- Slots – frames, nodes and switch boards.

The hardware objects are hierarchical. There is a system object (that cannot be added or deleted) that contains frame objects which contain slot objects (which represent nodes and switches). In checking for authorization, if a slot object exists the ACLs associated with it are used to determine access. If no slot object exists, ACLs for the frame object containing that slot will be checked if they exist. If the frame object doesn't exist, the system object is checked. The system object is the initial object in the system and must have a default ACL associated with it. If only one group of administrators will be responsible for the whole SP system, only the system object is required.

When an object is created the object will inherit the initial object or initial container ACLs of its container object (depending upon whether the object being created is a container itself). A slot is contained by its frame. If there is no existing object for its frame when a slot object is created, it will inherit the ACLs from the system object. If a group is specified on the **hmdceobj** command with the **−n** flag, that group is added to the default ACLs when the object is created.

Frame and slot objects are represented the same for the **hmdceobj** command as for the other hardmon commands (**hmmon** and **hmcmds**). Each frame consists of 18 slots, numbered 0 through 17, where slot 0 represents the frame itself, slot 17 can contain a switch, and slots 1 through 16 can contain thin or wide processing nodes. Wide nodes occupy two slots and are addressed by the odd slot number. In a switch only frame, slots 1 through 16 can contain switches; the switches occupy two slots and are addressed by the even slot number.

The hardware identifier operands are interpreted as slot ID specifications or node lists. A slot ID specification names one or more slots in one or more SP frames and it has the following form:

```
fidlist:sidlist
```

where:

```
fidlist = fval[,fval,...]
```

```
sidlist = sval[,sval,...]
```

An **fval** is a frame number or a range of frame numbers of the form a–b. An **sval** is a slot number from the set 0 through 17 or a range of slot numbers of the form a–b.

The form for a node list is:

```
nodlist = nval[,nval,...]
```

An **nval** is a node number or a range of node numbers of the form a–b.

The following are some examples of slot ID specifications.

1. To specify slot 1 in frames 1 through 10, enter:

   ```
   1-10:1
   ```

2. To specify frames 2, 4, 5, 6, and 7, enter:

   ```
   2,4-7:0
   ```

3. To specify slots 9 through 16 in frame 5, enter:

   ```
   5:9-16
   ```

   If frame 5 contained wide nodes, the even slot numbers are ignored.

4. To specify specifies slots 1, 12, 13, 14, 15, and 16 in each of frames 3 and 4, enter:

   ```
   3,4:1,12-16
   ```

5. To specify slot 17 in frame 4, enter:

   ```
   4:17
   ```

6. To specify the nodes in slots 1 through 16 of frame 2, enter:

   ```
   17-32
   ```

7. To specify the nodes in slot 1 of frame 1, slot 1 of frame 2 and slot 1 of frame 3, enter:

   ```
   1,17,33
   ```

8. To specify the node in slot 6 of frame 1, enter:

   ```
   6
   ```

Optionally, hardware identifiers can be provided in a file rather than as command operands. The file must contain one specification per line.

After creating objects, ACLs can be modified using the **dcecp** command or the RS6000/SP Security SMIT interface. A hardmon object will be represented by a string such as:

```
/.:/subsys/ssp/hardmon/{hostname}/{residual-part}
```

where **/.:/subsys/ssp/{hostname}/hardmon** is the CDS pathname for the hardmon daemon running on the SP system whose control workstation is identified by the specified DCE hostname. The hostname needs to be part of the database string since more than one SP system can be within the same DCE cell. The **{residual–part}** of the object string will be one of the following:

- system – represents the whole SP system
- hardmon – represents the hardmon daemon

- frame# – a number representing a frame
- frame#/slot# – a number representing a slot. Slot number can be in the range 0–17

Files

/spdata/sys1/spmon/hmdceacls

Standard Error

This command writes all error messages to standard error.

Exit Values

0 Indicates successful completion of the command.

1 Indicates that an error occurred during processing.

Security

This command uses the SP Hardware Monitor. To execute this command, users must be authorized to access the Hardware Monitor subsystem with administrative permission.

Restrictions

This command is only used if DCE is used as an authentication method.

Implementation Specifics

This command is part of the IBM Parallel System Support Program (PSSP) Licensed Program Product (LPP) ssp.clients file set.

Prerequisite Information

Refer to *PSSP: Administration Guide* for more information on DCE implementation on the SP.

Location

/usr/lpp/ssp/bin/hmdceobj

Related Information

Commands: **spacl**

Examples

1. To add an object for the switch in slot 17 in frame 1 with default permissions, enter:

   ```
   hmdceobj 1:17
   ```

2. To delete the object for the switch in slot 17 in frame 1, enter:

   ```
   hmdceobj -d 1:17
   ```

3. To add frame objects for frames 1 and 2 with monitor permission for DCE group monitor_only in addition to the default permissions, enter:

   ```
   hmdceobj -a -g monitor_only -p m 1,2:0
   ```

4. List all of the DCE objects defined in the hardware monitor's DCE database. Only existing objects are shown.

```
hmdceobj -q 0-3
```

Sample output:

```
system

hardmon

frame1

frame1/slot0

frame1/slot1

frame1/slot2

frame1/slot3
```

5. List whether DCE objects are defined for all slots, including default objects, with raw format output. The last number on each line is 0 for FALSE or 1 for TRUE. The hexadecimal values are variable identifiers.

```
hmdceobj -r 0-3
```

Sample output:

```
- - 0xa17d

- - 0xa17e

1 0 0xa175 1

1 0 0xa176 1

1 1 0xa176 1

1 2 0xa176 1

1 3 0xa176 1
```

6. List whether DCE objects are defined for all slots, including default objects, with symbolic format output. There will be a line for each frame and slot requested indicating whether the frame and slot have objects or not. The hexadecimal values are variable identifiers.

```
hmdceobj -s 0-3
```

Sample output:

```
| - - SystemHasObject  TRUE 0xa17d System Object Exists
| - - HaclmonHasObject TRUE 0xa17e Hardmon Object Exists
| 1 0 FrameHasObject   TRUE 0xa175 Frame Has Object
| 1 0 SlotHasObject    TRUE 0xa176 Slot Has Object
| 1 1 SlotHasObject    TRUE 0xa176 Slot Has Object
| 1 2 SlotHasObject    TRUE 0xa176 Slot Has Object
| 1 3 SlotHasObject    TRUE 0xa176 Slot Has Object
```

hmgetacls

Purpose

hmgetacls – Displays Hardware Monitor access permissions.

Syntax

hmgetacls [–**f** *file_name*] [*slot_spec*]

Flags

–**f** *file_name* Uses the *file_name* as the source of the slot ID specifications.

Operands

slot_spec
Specifies the addresses of the hardware components.

Description

The **hmgetacls** command Displays Hardware Monitor access permissions that are associated with the specified hardware, in conjunction with the configured authorization methods.

Files

/spdata/sys1/spmon/hmdceacls, **/spdata/sys1/spmon/hmacls**

Location

/usr/lpp/ssp/bin/hmgetacls

Related Information

Commands: **lsauthts**, **chauthts**, **hmckacls**, **hmdceobj**

Examples

In the following examples the same command (**hmgetacls 1:1-2**) is issued four times; once for each of the possible authorization method configurations. The command's target is slots 1 and 2 in frame 1. Slot 1 contains a wide node and therefore it is not possible for slot 2 to contain a node.

1. If the configured authentication method is DCE only, then the ACLs are applied at the slot level. In this example, slot 1 has **v**, **s**, **m**, and **u** permissions and slot 2 has none since slot 2 hardware does not exist:

```
[/]> hmgetacls 1:1-2

frame1/slot1      v  s  m  u

frame1/slot2      -  -  -  -
```

2. If the configured authentication method is COMPAT only, then the ACLs are applied at the frame level. Therefore, all slots within a frame will have the same permissions as the frame. Note that the **u** permission is not valid in COMPAT mode. It is a DCE only permission:

```
[/]> hmgetacls 1:1-2

frame1/slot1      v  s  m  -

frame1/slot2      v  s  m  -
```

3. If the configured authentication method is both DCE and COMPAT then the ACLs that are applied is the result of performing an "or" operation on both DCE and COMPAT permissions:

```
[/]> hmgetacls 1:1-2

frame1/slot1      v  s  m  u

frame1/slot2      v  s  m  -
```

4. If the configured authentication method is neither DCE nor COMPAT (there is no configured method), then access is granted based on whether the user is root and on the control workstation.

   If the user is root and on the control workstation, the output shows that the user can perform all of the actions:

```
[/]> hmgetacls 1:1-2

frame1/slot1      v  s  m  u

frame1/slot2      v  s  m  u
```

hmmon

Purpose

hmmon – Monitors the state of the SP hardware.

Syntax

hmmon [–**G**] [–**q**] [–**Q**] [–**r** | –**s**] [–**v** *var_nlist*]
[–**f** *file_name* | *slot_spec* ...]

hmmon –**V**

Flags

–**G** Specifies Global mode. With this flag, all hardware can be specified.

–**q** Displays the current state information prior to displaying changed state.

–**Q** Displays only the current state information and exits.

–**r** Displays the output in *raw* format.

–**s** Displays the output in *symbolic* format.

–**v** *var_nlist* Limits output to that of the state variables specified by *var_nlist*, a comma separated list of symbolic variable names. This list cannot contain blanks. Use the –**V** flag for a list of possible values.

–**V** Displays a descriptive list of symbolic variable names and variable indexes, and exits.

–**f** *file_name*
Uses the file *file_name* as the source of slot ID specifications.

Operands

slot_spec Displays the addresses of hardware components.

Description

Use this command to monitor the state of the SP hardware contained in one or more SP frames. Each frame consists of 18 slots, numbered 0 through 17, where slot 0 represents the frame itself, slot 17 can contain a switch and slots 1 through 16 can contain thin or wide processing nodes. Wide nodes occupy two slots and are addressed by the odd slot number. In a switch only frame, slots 1 through 16 can contain switches; the switches occupy two slots and are addressed by the even slot number.

With no flags and operands, the command prints to standard output descriptive text of all hardware state changes in the current system partition as they occur, from the time the command is invoked. The command does not terminate, unless the –**Q** flag or the –**V** flag is specified, and must be interrupted by the user. To monitor all of the hardware in the SP system, the –**G** flag must be specified. Note that the switches and the frames themselves are not contained in any system partition.

When one or more *slot_spec* operands are present, each operand is interpreted as a slot ID specification. A slot ID specification names one or more slots in one or more SP frames and it has either of two forms:

```
fidlist:[sidlist]   or   nodlist
```

where:

fidlist = fval[,fval,...]

sidlist = sval[,sval,...]

nodlist = nval[,nval,...]

The first form specifies frame numbers and slot numbers. The second form specifies node numbers. A *fval* is a frame number or a range of frame numbers of the form *a–b*. A *sval* is a slot number from the set 0 through 17 or a range of slot numbers of the form *a–b*. An *nval* is a node number or a range of node numbers of the form *a–b*. If a *sidlist* is not specified, all hardware in the frames specified by the *fidlist* is monitored.

The relationship of node numbers to frame and slot numbers is given by the following formula:

```
node_number = ((frame_number – 1) × 16) + slot_number
```

Note: The node numbers can only be used to specify slots 1 through 16 of any frame.

The following are some examples of slot ID specifications.

To specify all hardware in frames 1 through 10, enter:

```
1-10:
```

To specify frames 2, 4, 5, 6, and 7, enter:

```
2,4-7:0
```

To specify slots 9 through 16 in frame 5, enter:

```
5:9-16
```

If frame 5 contained wide nodes, the even slot numbers are ignored.

To specify slots 1, 12, 13, 14, 15, and 16 in each of frames 3 and 4, enter:

```
3,4:1,12-16
```

To specify slot 17 in frame 4, enter:

```
4:17
```

To specify the nodes in slots 1 through 16 of frame 2, enter:

```
17-32
```

To specify the nodes in slot 1 of frame 1, slot 1 of frame 2 and slot 1 of frame 3, enter:

```
1,17,33
```

To specify the node in slot 6 of frame 1, enter:

```
6
```

Optionally, slot ID specifications may be provided in a file rather than as command operands. The file must contain one specification per line. When slot ID specifications are provided to the command, only the hardware named by the specifications is monitored. Furthermore, of the hardware named by these specifications, only that which is located in the current system partition is monitored. To monitor hardware not contained in the current system partition, the −**G** flag must be specified. If the −**G** flag is not specified and the slot ID specifications name no hardware in the current system partition, an error message is displayed.

The default output displays hardware state information on a slot-by-slot basis. The state information for each slot is captioned by its frame ID and slot ID and consists of two columns. Each column contains state variable information, one variable per line. Each variable is displayed as descriptive text and a value. Boolean values are displayed as TRUE or FALSE. Integer values are displayed in hexadecimal.

The command provides two other output formats, raw and symbolic. Both write the information for one state variable per line. The *raw* format consists of four fields separated by white space as follows:

Field 1 Contains the frame ID.

Field 2 Contains the slot ID.

Field 3 Contains the variable ID in hexadecimal.

Field 4 Contains the variable value, as received from the hardware, in decimal.

The *symbolic* format consists of six fields separated by white space as follows:

Field 1 Contains the frame ID.

Field 2 Contains the slot ID.

Field 3 Contains the symbolic name of the state variable.

Field 4 Contains the variable value. Booleans are displayed as TRUE or FALSE. Integers are displayed as decimal values or floating point values, as appropriate to the definition of the variable.

Field 5 Contains the variable ID in hexadecimal.

Field 6 Contains the descriptive text for the variable. This is the same text that is displayed in the default output. Thus, "field" 6 contains embedded white space.

The alternative output formats are suitable for input to post-processing programs, such as **awk** or scripts.

Output in any format can be limited to display only information from the specified hardware that corresponds to a list of state variables supplied to the command with the −**v** flag.

The user can monitor nonexistent nodes in an existing frame to detect when a node is added while the system is up and running. No information is returned for nonexistent nodes when the −**q** or −**Q** flag is specified.

The **hmmon** command fails if any of the hardware targeted is not supported by the version of PSSP on the machine where the command was issued.

Security

To execute the **hmmon** command, the user must be authorized to access the Hardware Monitor subsystem and must be granted Monitor permission for the hardware objects (frames, slots) specified in the command. State information is not returned for hardware objects for which the user does not have Monitor permission.

Location

/usr/lpp/ssp/bin/hmmon

Related Information

Commands: **hmcmds**

Files: **/spdata/sys1/spmon/hmdceacls**, **/spdata/sys1/spmon/hmacls**

Examples

The following is an example of default output from **hmmon** −**G** −**Q 1:0,1**. The command returns similar output, depending on your system configuration.

```
frame 001, slot 00:

 node 01 I2C not responding     FALSE

 node 02 I2C not responding     TRUE

 node 03 I2C not responding     FALSE

 node 04 I2C not responding     TRUE

 switch I2C not responding      FALSE

 node 01 serial link open       TRUE

 node 02 serial link open       FALSE

 node 03 serial link open       TRUE

 frame LED 1 (green)            0x0001

 frame LED 2 (green)            0x0001

 frame LED 3 (yellow)           0x0000

 frame LED 4 (yellow)           0x0000

 AC-DC section A power off      FALSE

 AC-DC section B power off      FALSE

 AC-DC section C power off      FALSE

 AC-DC section D power off      FALSE

 supervisor timer ticks         0x88f2

 +48 voltage                    0x0078

 temperature                    0x0036

 supervisor serial number       0x1234

 supervisor type                0x0011

 supervisor code version        0x5ff5
```

```
frame 001, slot 01:

 serial 1 DTR asserted         TRUE

 -12 volt low warning          TRUE

 -12 volt low shutdown         FALSE

 -12 volt high warning         TRUE

 +4 volt low shutdown          FALSE

 +4 volt high warning          TRUE

 fan 1 shutdown                FALSE

 fan 2 warning                 TRUE

 DC-DC power on > 10 secs      TRUE

 +5 DC-DC output good          TRUE

 7 segment display flashing    FALSE

 node/switch LED 1 (green)     0x0001

 reset button depressed        FALSE
 serial link open              TRUE

 diagnosis return code         0x00dd

 7 segment LED A               0x00ff

 +5 I/O voltage                0x007f

 +12 voltage                   0x0096
```

The following is an example of *raw* output from **hmmon −G −Q −r 1:0**. The command returns similar output, depending on your system configuration.

```
1 0 0x880f 32

1 0 0x881c 0

1 0 0x881d 4

1 0 0x8834 54

1 0 0x8839 4660

1 0 0x883a 17

1 0 0x88a8 1

1 1 0x9097 16

1 1 0x9098 0

1 1 0x9047 1

1 1 0x909d 128

1 1 0x9023 221

1 1 0x90a1 255

1 1 0x90a2 127

1 1 0x903b 24565
```

The following is an example of *symbolic* output from **hmmon −G −Q −s 1:0**. The command returns similar output, depending on your system configuration.

```
1  0  nodefail1          FALSE    0x8802  node 01 I2C not responding

1  0  nodeLinkOpen1      TRUE     0x8813  node 01 serial link open

1  0  frACLED                 1   0x8824  frame LED 1 (green)

1  0  frNodeComm              0   0x8827  frame LED 4 (yellow)

1  0  frPowerOff_B       FALSE    0x882d  AC-DC section B power off

1  0  timeTicks           34881   0x8830  supervisor timer ticks

1  0  voltP48            46.800   0x8831  +48 voltage

1  0  type                   17   0x883a  supervisor type

1  0  codeVersion         24565   0x883b  supervisor code version

1  0  controllerResponds TRUE     0x88a8  Frame responding to polls

1  0  rs232DCD           TRUE     0x88a9  RS232 link DCD active

1  0  rs232CTS           TRUE     0x88aa  RS232 link CTS active

1  1  fanfail2           FALSE    0x9050  fan 2 shutdown

1  1  nodePowerOn10Sec   TRUE     0x904b  DC-DC power on > 10 secs

1  1  P5DCok             TRUE     0x9097  +5 DC-DC output good

1  1  powerLED                1   0x9047  node/switch LED 1 (green)

1  1  envLED                  0   0x9048  node/switch LED 2 (yellow)

1  1  keyModeSwitch           0   0x909b  key switch

1  1  serialLinkOpen     TRUE     0x909d  serial link open

1  1  LED7SegA              255   0x909f  7 segment LED A

1  1  voltP5i             4.978   0x90a2  +5 I/O voltage
```

The raw and symbolic formats output by the **hmmon** command contain the variable ID of each state variable. Refer to Appendix D in *PSSP: Administration Guide*.

hmreinit

Purpose

hmreinit – Stops and starts the Hardware Monitor daemon and modifies the System Data Repository (SDR) as necessary.

Syntax

hmreinit

Flags

None.

Operands

None.

Description

Use this command to reinitialize the Hardware Monitor daemon when changes to the SP system occur. When the daemon is restarted, **splogd** will invoke **SDR_config** which will update the SDR to match the current hardware configuration.

Standard Error

This command writes error messages (as necessary) to standard error.

Exit Values

0 Indicates the successful completion of the command.

| **1, 2, 3** Indicates an error occurred. Check error messages for information.

Security

| You must have root privilege and write access to the SDR to run this command.

Implementation Specifics

This command is part of the Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Location

/usr/lpp/ssp/install/bin/hmreinit

Related Information

Commands: **SDR_config**, **spframe**, **splogd**

For additional information, refer to the “Reconfiguring the IBM RS/6000 SP System” chapter in *PSSP: Installation and Migration Guide*.

Examples

To stop and restart the Hardware Monitor daemon, enter:

```
hmreinit
```

You should receive messages similar to the following:

```
0513-044 The stop of the splogd Subsystem was completed successfully.

0513-044 The stop of the hardmon Subsystem was completed successfully.

0513-059 The hardmon Subsystem has been started. Subsystem PID is 22746.

0513-059 The splogd Subsystem has been started. Subsystem PID is 28440.

SDR_config: SDR_config completed successfully.
```

hostlist

Purpose

hostlist – Lists SP host names to standard output based on criteria.

Syntax

hostlist [–**s** *framerange*:*slotrange*] [–**f** *file_name*] [–**a**] [–**G**]
[–**n** *noderange*] [–**w** *host_name*,*host_name*, ...]
[–**e** *host_name*,*host_name*, ...] [–**v**] [–**d** | –**l**] [–**r**]
[–**N** *node_group*,*node_group*, ...]

Flags

–s Specifies a range of frames and a range of slots on each of the frames. Ranges are specified as in 1–3, meaning 1 through 3 inclusive, and as 1,3,15, meaning 1, 3, and 15. Ranges can incorporate both styles as in 1–10,15. So, 1–3,5:1–2,4 would refer to slots 1,2 and 4 on each of the frames 1,2,3, and 5. If a node occupies more than one slot, referring to either or both of the slots refers to the node.

–f Specifies the file name of a working collective file as in the **dsh** working collective, containing a host name on each line. This can be in the format of a Parallel Operating Environment (POE) **host.list** file.

–a Specifies that the System Data Repository (SDR) **initial_hostname** field for all nodes in the current system partition be written to standard output. For each node, this corresponds to what the **hostname** command returns on the node.

–G Changes the scope of the arguments associated with the –**a**, –**n**, –**s**, and –**N** options from the current system partition to the SP system.

–n Specifies that all nodes in a *noderange* are written. The range specification has syntax similar to that of frame or slot ranges. Nodes are numbered starting with 1, for frame 1 slot 1, up to the number of slots on the system (note that a node number can refer to an empty slot). A *noderange* can span frames (for example, 1–4,17–50) would refer to all nodes occupying slots 1–4 on frame 1 and 1–16 on frames 2 and 3, and slots 1 and 2 on frame 4.

–w Specifies a list of host names, separated by commas, to include in the working collective. Both this flag and the **a** flag can be included on the same command line. Duplicate host names are only included once in the working collective.

–e Specifies an exclusion list. Comma-separated host names specified are not written to standard output.

–v Specifies that only nodes that are responding according to the SDR have their host names written.

–d Specifies that IP addresses are returned as output.

–l Specifies that long host names be written. (This is lowercase **l**, as in **l**ist.)

–r Specifies a restriction to write host names for only those nodes that have exactly the same node number or starting slot specified by the search argument. For example, if a "-n" value corresponds to the second slot of a wide node, and the "-r" flag is used, then a warning message is written instead of the host name for the first slot of the wide node.

–N Specifies a list of node groups. Each node group is resolved into nodes. The host names of these nodes are added to the host list. If **–G** is supplied, a global node group is used. Otherwise, a partitioned-bound node group is used.

Operands

None.

Description

The **hostlist** command writes SP host names to standard output. The arguments to the command indicate the host names to be written. More than one flag can be specified, in which case, the hosts indicated by all the flags are written.

If no arguments are specified, **hostlist** writes the contents of a file specified by the WCOLL environment variable. If the WCOLL environment variable does not exist, the MP_HOSTFILE environment variable is used as the name of a POE host file to use for input. Finally, **./host.list** is tried. If none of these steps are successful, an error has occurred. The input file is in **dsh**-working-collective-file or POE-host-list-file format. Node pool specifications in POE host files are not supported.

Files

working collective file
See the **dsh** command.

POE host.list file
See *Parallel Environment for AIX: Operation and Use* documentation.

Related Information

Commands: **dsh**, **sysctl**

Examples

1. To create a working collective file of all nodes in the system partition that are responding, except for badhost, enter:
```
hostlist -av -e badhost > ./working
```
2. To run a program on the nodes on slot 1 of each of 4 frames, enter:
```
hostlist -s 1-4:1 | dsh -w - program
```
3. To run a program on the nodes on all slots for frame 1 and slots 1-3 for frame 3, as well as on host **otherone**, enter:
```
hostlist -n 1-16,33-35 -w otherone | dsh -w - program
```
4. To run a Sysctl application on all the nodes in the WCOLL file **./wcoll:**, enter:
```
export WCOLL=./wcoll

hostlist | sysctl -c - sysctl_app args
```

hr Script

Purpose

hr – Controls the host_responds monitor daemon, **hrd**, on the control workstation.

Syntax

hr [–**spname** *syspar_name*]

{ [**start** | **resume**] | [**stop** | **quiesce**] | **reset** |

[**query** | **qall** | **qsrc**] | **refresh** | **mksrc** *optional_flags* | **rmsrc** | **clean** | **restore** |

[**debug** | **debug off**] | [**trace on** | **trace off**] }

Flags

–**spname** *syspar_name*
Executes the command for the system partition specified by the *syspar_name* operand. If this flag is not specified, the name of the system partition given by the value of the SP_NAME variable is used.

Operands

start | resume
Starts the **hrd** daemon.

stop | quiesce
Stops the **hrd** daemon.

reset Stops and restarts the **hrd** daemon.

query Queries the daemon for status. The response to the **query** includes **hrd**-specific information.

qall Performs the **query** function for each defined partition.

qsrc Displays a subsystem definition for a partition.

refresh Uses the refresh command to request a daemon refresh.

mksrc *optional_flags*
Uses the **mkssys** command to create an SRC subsystem object. Additional flags for the command may be specified.

rmsrc Uses the **rmssys** command to remove an SRC subsystem object.

clean Removes all entries for the subsystem for all system partitions.

restore Synchronizes the running daemons with the information in the System Data Repository (SDR). This operand removes all entries for the subsystem, creates new entries based on information in the SDR, and starts the subsystems.

[debug | debug off]
Turns debugging on or off.

[trace on | trace off]
Turns additional tracing on or off.

Description

Use this command to control the operation of **hrd**, the host_responds daemon on the control workstation within a system partition. The **hrd** daemon receives information from the Event Management subsystem about nodes that are reachable through SP ethernet within a system partition. The **hrd** daemon uses this information to update the SDR host_responds class.

The **hr** script is not normally executed from the command line. It is normally called by the **hrctrl** command, which is in turn called by the **syspar_ctrl** command during installation of the system, and partitioning or repartitioning of the system.

The **hrd** daemon is initially started on the control workstation with the System Resource Controller (SRC). It is respawned automatically if the **hrd** daemon is unsuccessful. The SP_NAME environment variable causes selection of the correct daemon.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

The “Starting Up and Shutting Down the SP System” chapter and “The System Data Repository” appendix in *PSSP: Administration Guide*

AIX Version 4 Commands Reference

Information about the System Resource Controller (SRC) in *AIX Version 4 General Programming Concepts: Writing and Debugging Programs*

Location

/usr/lpp/ssp/bin/hr

Related Information

Commands: **hrctrl**, **lssrc**, **startsrc**, **stopsrc**, **syspar_ctrl**

Examples

See the **hrctrl** command.

hrctrl Script

Purpose

hrctrl – A script that controls the Host_Responds subsystem.

Syntax

hrctrl { **–a** | **–s** | **–k** | **–d** | **–c** | **–t** | **–o** | **–r** | **–h** }

Flags

–a Adds the subsystem.

–s Starts the subsystem.

–k Stops the subsystem.

–d Deletes the subsystem.

–c Cleans the subsystems, that is, delete them from all system partitions.

–t Turns tracing on for the subsystem.

–o Turns tracing off for the subsystem.

–r Refreshes the subsystem.

–h Displays usage information.

Operands

None.

Description

The Host_Responds subsystem provides to other PSSP subsystems information about the state of the nodes on the IBM RS/6000 SP.

The **hrctrl** control script controls the operation of the Host_Responds subsystem. The subsystem is under the control of the System Resource Controller (SRC) and belongs to a subsystem group called **hr**. Associated with each subsystem is a daemon and a script that configures and starts the daemon.

An instance of the Host_Responds subsystem executes on the control workstation for every system partition. Because Host_Responds provides its services within the scope of a system partition, its subsystem is said to be system partition-sensitive. This control script operates in a manner similar to the control scripts of other system partition-sensitive subsystems. The script should be issued on the control workstation. If it is issued on a node, it has no effect.

From an operational point of view, the Host_Responds subsystem group is organized as follows:

Subsystem Host_Responds

Subsystem Group **hr**

SRC Subsystem	**hr**
	The **hr** subsystem is associated with the **hrd** daemon and the **hr** script. The **hr** script configures and starts the **hrd** daemon.
	On the control workstation, there are multiple instances of each subsystem, one for each system partition. Accordingly, the subsystem names on the control workstation have the system partition name appended to them. For example, for system partitions named **sp_prod** and **sp_test**, the subsystems on the control workstation are named **hr.sp_prod** and **hr.sp_test**.
	The subsystem does not run on the nodes.
Daemons	**hrd**
	The **hrd** daemon provides the Host_Responds services. The **hr** script configures and starts the **hrd** daemon.

The **hrctrl** script is not normally executed from the command line. It is normally called by the **syspar_ctrl** command during installation of the system, and partitioning or repartitioning of the system.

The **hrctrl** script provides a variety of controls for operating the Host_Responds subsystem:

- Adding, starting, stopping, and deleting the subsystem
- Cleaning up the subsystems, that is, deleting them from all system partitions
- Turning tracing on and off
- Refreshing the subsystem.

Before performing any of these functions, the script obtains the node number (using the **node_number**) command. If the node number is not zero, the control script is running on a node and it exits immediately. Otherwise, it is executing on the control workstation and it calls the **hr** script with an operand that specifies the action to be performed.

Adding the Subsystem

When the **−a** flag is specified, the control script uses the **hr** command with the **mksrc** operand to add the Host_Responds subsystem to the SRC.

Starting the Subsystem

When the **−s** flag is specified, the control script uses the **hr** command with the **start** operand to start the Host_Responds subsystem, **hr**.

Stopping the Subsystem

When the **−k** flag is specified, the control script uses the **hr** command with the **stop** operand to stop the Host_Responds subsystem, **hr**.

Deleting the Subsystem

When the –**d** flag is specified, the control script uses the **hr** command with the **rmsrc** operand to remove the Host_Responds subsystem from the SRC.

Cleaning up the Subsystems

When the –**c** flag is specified, the control script uses the **hr** command with the **clean** operand to stop and remove the Host_Responds subsystems for all system partitions from the SRC.

Turning Tracing On

When the –**t** flag is specified, the control script turns tracing on for the **hrd** daemon, using the **hr** command with the **trace on** operand.

Turning Tracing Off

When the –**o** flag is specified, the control script turns tracing off (returns it to its default level) for the **hrd** daemon, using the **hr** command with the **trace off** operand.

Refreshing the Subsystem

When the –**r** flag is specified, the control script refreshes the subsystem, using the **hr refresh** command.

Standard Error

This command writes error messages (as necessary) to standard error.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error occurred.

Security

| You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

AIX Version 4 Commands Reference

Information about the System Resource Controller (SRC) in *AIX Version 4 General Programming Concepts: Writing and Debugging Programs*

Location

/usr/lpp/ssp/bin/hrctrl

Related Information

Commands: **hr**, **lssrc**, **startsrc**, **stopsrc**, **syspar_ctrl**

Examples

1. To add the Host_Responds subsystem to the SRC in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   hrctrl -a
   ```

2. To start the Host_Responds subsystem in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   hrctrl -s
   ```

3. To stop the Host_Responds subsystem in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   hrctrl -k
   ```

4. To delete the Host_Responds subsystem from the SRC in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   hrctrl -d
   ```

5. To clean up the Host_Responds subsystem on all system partitions, enter:

   ```
   hrctrl -c
   ```

6. To turn tracing on for the Host_Responds daemon in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   hrctrl -t
   ```

7. To turn tracing off for the Host_Responds daemon in the current system partition, set the SP_NAME environment variable to the appropriate system partition name and enter:

   ```
   hrctrl -o
   ```

8. To display the status of all of the subsystems in the Host_Responds SRC group, enter:

   ```
   lssrc -g hr
   ```

9. To display the status of an individual Host_Responds subsystem, enter:

   lssrc -s *subsystem_name*

10. To display detailed status about an individual Host_Responds subsystem, enter:

    lssrc -l -s *subsystem_name*

In response, the system returns information that includes the running status of the subsystem and the status of the nodes within the system partition.

11. To display the status of all of the daemons under SRC control, enter:

```
lssrc -a
```

hsdatalst

Purpose

hsdatalst – Displays hashed shared disk information for the virtual shared disk from the System Data Repository (SDR).

Syntax

hsdatalst [–**G**]

Flags

–**G** Displays information for all system partitions on the SP, not only the current system partition.

Operands

None.

Description

This command is used to display defined hashed shared disk information in the system.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit list_vsd
```

and select the List Defined Hashed Shared Disk option.

Prerequisite Information

PSSP: Managing Shared Disks

Location

/usr/lpp/csd/bin/hsdatalst

Related Information

Commands: **defhsd**, **undefhsd**, **updatehsd**

Examples

To display SDR hashed shared disk data, enter:

```
hsdatalst
```

which produces output similar to:

```
                     HSD Table

minor option        stripeSZ #vsds hsd_name vsd_name  size_in_MB

---- ------------- -------- ----- -----------------------------

1     protect_lvcb  32768     2     HsD     HsD1n13   512

1     protect_lvcb  32768     2     HsD     HsD1n14   512

2     protect_lvcb  65536     4     myhsd   vsd1n1    2148

2     protect_lvcb  65536     4     myhsd   vsd1n2    2148

2     protect_lvcb  65536     4     myhsd   vsd1n3    2148

2     protect_lvcb  65536     4     myhsd   vsd1n4    2148
```

hsdvts

Purpose

hsdvts – Verifies that a hashed shared disk for a virtual shared disk has been correctly configured and works.

Attention

Data on *hsd_name* will be overwritten and, therefore, destroyed. Use this command after you have defined your hashed shared disks, virtual shared disks, and logical volumes, but **before** you have loaded your application data onto any of them.

Syntax

hsdvts *hsd_name*

Flags

None.

Operands

hsd_name The name of the hashed shared disk you want verified. **Warning:** Data on *vsd_name* will be overwritten and, therefore, destroyed.

Description

This command writes **/unix** to *hsd_name*, reads it from *hsd_name* to a temporary file, and compares the temporary file to the original to make sure the I/O was successful. If the files compare exactly, the test was successful.

hsdvts writes to the raw *hsd_name* device **/dev/r***hsd_name*. Since raw devices can only be written in multiples of 512-sized blocks, **hsdvts** determines the number of full 512-byte blocks in **/unix** file, and writes that number to *hsd_name* via **dd** command. It makes a copy of **/unix** that contains this number of 512-byte blocks for comparison to the copy read from *hsd_name*. The **dd** command is used for all copy operations.

Prerequisite Information

PSSP: Managing Shared Disks

Location

/usr/lpp/csd/bin/hsdvts

Related Information

Commands: **cfghsd**, **cfgvsd**, **dd**, **defhsd**, **startvsd**

Examples

To verify that hashed shared disk hsd1 has been correctly configured and works, enter:

```
hsdvts hsd1
```

ifconfig

Purpose

ifconfig – Configures or displays network interface parameters for a network using TCP/IP.

Syntax

ifconfig *interface* [*address_family* [*address* [*destination_address*]] [*parameter*...]]

Flags

None.

Operands

address
Specifies the network address for the network interface. For the inet family, the *address* operand is either a host name, or an Internet address in the standard dotted decimal notation.

address_family
Specifies which network address family to change. The inet and ns address families are currently supported. This operand defaults to the inet address family.

destination_address
Specifies the address of the correspondent on the remote end of a point-to-point link.

interface
Specifies the network interface configuration values to show or change. You must specify an interface with the *interface* operand when you use the **ifconfig** command. Abbreviations for the interfaces include:

en Standard Ethernet (inet, xns)
et IEEE 802.3 Ethernet (inet, xns)
tr Token ring (inet, xns)
xt X.25 (inet)
sl Serial line IP (inet)
lo Loopback (inet)
op Serial (inet)
css Scalable POWERparallel Switch (SP Switch)

Include a numeral after the abbreviation to identify the specific interface (for example, tr0).

parameter
Allows the following parameter values:

alias
Establishes an additional network address for the interface. When changing network numbers, this is useful for accepting packets addressed to the old interface.

allcast
Sets the token-ring interface to broadcast to all rings on the network.

–**allcast**
Confines the token-ring interface to broadcast only to the local ring.

arp Enables the **ifconfig** command to use the Address Resolution Protocol (ARP) in mapping between network-level addresses and link-level addresses. This flag is in effect by default.

−**arp** Disables the use of the Address Resolution Protocol.

authority Reserved.

bridge Reserved.

−**bridge** Reserved.

broadcast_address
(inet only). Specifies the address to use to broadcast to the network. The default broadcast address has a host part of all 1's (ones).

debug Enables driver-dependent debug code.

−**debug** Disables driver-dependent debug code.

delete Removes the specified network address. This is used when an alias is incorrectly specified or when it is no longer needed. Incorrectly setting ns addresses have the side effect of specifying the host portion of the network address. Removing all ns addresses allows you to respecify the host portion.

detach Removes an interface from the network interface list. If the last interface is detached, the network interface driver code is unloaded.

down Marks an interface as inactive (down), which keeps the system from trying to transmit messages through that interface. If possible, the **ifconfig** command also resets the interface to disable reception of messages. Routes that use the interface, however, are not automatically disabled.

hwloop Enables hardware loopback. The hardware loopback specifies that locally-addressed packets handled by an interface should be sent out using the associated adapter.

−**hwloop** Disables hardware loopback. The hardware loopback specifies that locally-addressed packets handled by an interface should be sent out using the associated adapter.

ipdst Specifies an Internet host willing to receive IP packets encapsulating ns packets bound for a remote network. An apparent point-to-point link is constructed, and the specified address is taken as the ns address and network of the destination.

metric_number
Sets the routing metric of the interface to the value specified by the *number* variable. The default is 0. The routing metric is used by the routing protocol (the routed daemon). Higher metrics have the effect of

making a route less favorable. Metrics are counted as addition hops to the destination network or host.

mtu_value
: Sets the maximum IP packet size for this system. The *value* variable can be any number from 60 through 65520, depending on the network interface. See “Understanding Automatic Configuration of Network Interfaces” in *AIX Version 4 System Management Guide: Communications and Networks* for maximum transmission unit (MTU) values by interface.

netmask_mask
: Specifies how much of the address to reserve for subdividing networks into subnetworks. This parameter can only be used with an address family of inet.

  The *mask* variable includes both the network part of the local address and the subnet part, which is taken from the host field of the address. The *mask* can be specified as a single hexadecimal number beginning with 0x, in standard Internet dotted decimal notation, or beginning with a name or alias that is listed in the **/etc/networks** file.

  The *mask* contains 1's (ones) for the bit positions in the 32-bit address that are reserved for the network and subnet parts, and 0's (zeros) for the bit positions that specify the host. The *mask* should contain at least the standard network portion, and the subnet segment should be contiguous with the network segment.

offset
: Used by the CSS/IP for static IP address translation only.

  Note: If the ARP is enable, offset is not used.

TB0/TB2
: Indicates to the CSS/IP whether it is running over TB0 or TB2 adapter interface. The default is TB2 adapter.

security
: Reserved.

snap
: Reserved.

–**snap**
: Reserved.

up
: Marks an interface as active (up). This parameter is used automatically when setting the first address for an interface. It can also be used to enable an interface after an **ifconfig down** command.

Description

The **ifconfig** command has been modified to add support for the switch. This command is valid only on an SP system.

The **ifconfig** command can be used from the command line either to assign an address to a network interface, or to configure or display the current network interface configuration information. The **ifconfig** command must be used at system start up to define the network address of each interface present on a machine. It can also be used at a later time to redefine an interface's address or other operating parameters. The network interface configuration is held on the running system and must be reset at each system restart.

An *interface* can receive transmissions in differing protocols, each of which may require separate naming schemes. It is necessary to specify the *address_family* parameter, which can change the interpretation of the remaining parameters. The address families currently supported are inet and ns.

For the DARPA Internet family, inet, the address is either a host name present in the host name database, that is, the **/etc/hosts** file, or a DARPA Internet address expressed in the Internet standard dotted decimal notation.

For the Xerox Network Systems (XNS) family, ns, addresses are net:a.b.c.d.e.f., where net is the assigned network number (in decimal), and each of the six bytes of the host number, a through f, are specified in hexadecimal. The host number can be omitted on 10-Mbps Ethernet interfaces, which use the hardware physical address, and on interfaces other than the first interface.

While any user can query the status of a network interface, only a user who has administrative authority can modify the configuration of those interfaces.

Related Information

AIX Command: **netstat**

AIX Files: **/etc/host**, **/etc/networks**

Refer to *PSSP: Administration Guide* for additional information on the SP Switch.

Refer to *AIX Version 4 System Management Guide: Communications and Networks* for additional information on TCP/IP protocols.

Refer to *AIX Version 4 General Programming Concepts: Writing and Debugging Programs* for an overview on Xerox Network Systems (XNS).

Location

/usr/lpp/ssp/css/ifconfig

Examples

The following are examples using the **ifconfig** command on a TCP/IP network and an XNS network, respectively:

Inet Examples

1. To query the status of a serial line IP interface, enter:

   ```
   ifconfig sl1
   ```

   In this example, the interface to be queried is sl1. The result of the command looks similar to the following:

   ```
   sl1: flags=51<UP,POINTOPOINT,RUNNING>

        inet 192.9.201.3 --> 192.9.354.7 netmask ffffff00
   ```

2. To configure the local loopback interface, enter:

   ```
   ifconfig lo0 inet 127.0.0.1 up
   ```

3. To mark the local token-ring interface as down, enter:

   ```
   ifconfig tr0 inet down
   ```

   In this example, the interface to be marked is token0.

   Note: Only a user with root user authority can modify the configuration of a network interface.

4. To specify an alias, enter:

   ```
   ifconfig css0 inet 127.0.0.1 netmask 255.255.255.0 alias
   ```

XNS Examples

1. To configure a standard Ethernet-type interface for XNS, enter:

   ```
   ifconfig en0 ns 110:02.60.8c.2c.a4.98 up
   ```

   In this example, ns is the XNS address family, 110 is the network number and 02.60.8c.2c.a4.98 is the host number, which is the Ethernet address unique to each individual interface. Specify the host number when there are multiple Ethernet hardware interfaces, as the default may not correspond to the proper interface. The Ethernet address can be obtained by the commands:

   ```
   ifconfig en0

   netstat -v
   ```

   The XNS address can be represented by several means, as can be seen in the following examples:

```
123#9.89.3c.90.45.56
```

```
5-124#123-456-900-455-749
```

```
0x45:0x9893c9045569:90
```

```
0456:9893c9045569H
```

The first example is in decimal format, and the second example, using minus signs, is separated into groups of three digits each. The 0x and H examples are in hexadecimal format. Finally, the 0 in front of the last example indicates that the number is in octal format.

2. To configure an IEEE Ethernet 802.3-type interface for XNS, enter:

   ```
   ifconfig et0 ns 120:02.60.8c.2c.a4.98 up
   ```

   The en0 and et0 interfaces are considered as separate interfaces even though the same Ethernet adapter is used. Two separate networks can be defined and used at the same time as long as they have separate network numbers. Multiple Ethernet adapters are supported.

   Note: The host number should correspond to the Ethernet address on the hardware adapter. A system can have multiple host numbers.

3. To configure an Internet encapsulation XNS interface, enter:

   ```
   ifconfig en0 inet 11.0.0.1 up

   ifconfig en0 ns 110:02.60.8c.2c.a4.98 up

   ifconfig en0 ns 130:02.60.8c.34.56.78 ipdst 11.0.0.10
   ```

   The first command brings up the Internet with the inet address 11.0.0.1. The second command configures the en0 interface to be network 110 and host number 02.60.8c.2c.a4.98 in the ns address family. This defines the host number for use when the XNS packet is encapsulated within the Internet packet. The last command defines network 130, host number 02.60.8c.34.56.78, and destination Internet address 11.0.0.10. This last entry creates a new network interface, nsip. Use the **netstat -i** command for information about this interface.

install_cw

Purpose

install_cw – Completes the installation of system support programs in the control workstation.

Syntax

install_cw

Flags

None.

Operands

None.

Description

Use this command at installation to perform the following tasks:

- Installs IBM Parallel System Support Programs for AIX (PSSP) SMIT Panels
- Starts and configures the System Data Repository (SDR)
- Sets the node number to 0 in the Object Data Management (ODM) table
- Starts and configures PSSP daemons
- Establishes default network performance tuning parameters for the SP nodes by copying **/usr/lpp/ssp/install/config/tuning.default** to **/tftpboot/tuning.cust** (only if the **tuning.cust** file does not exist).

Security

You must have root privilege to run this command. You must also have SDR administrator and write access to run this command.

Location

/usr/lpp/ssp/bin/install_cw

Examples

To complete system support programs installation on the control workstation, enter:

```
install_cw
```

install_hacws

Purpose

install_hacws – Creates and configures a High Availability Control Workstation (HACWS) configuration from a regular control workstation configuration.

Syntax

install_hacws −**p** *host_name* −**b** *host_name* [−**s**]

Flags

−**p** Specifies the host name of the primary control workstation. The host name is the name that is set in the kernel and identifies the physical machine. It is also required that this name have a route defined to a network adapter on the primary control workstation. This option is required.

−**b** Specifies the host name of the backup control workstation. The host name is the name that is set in the kernel and identifies the physical machine. It is also required that this name have a route defined to a network adapter on the backup control workstation. This option is required.

−**s** Invokes the command on both the primary and the backup control workstations.

Operands

None.

Description

Use this command to perform configuration and installation tasks on HACWS. This command is used instead of **install_cw** once the configuration has been made an HACWS configuration. This command is valid only when issued on the control workstation. When the command is executed and the calling process is not on a control workstation, an error occurs.

Note: The **install_hacws** command permanently alters a control workstation to an HACWS. The only way to go back to a single control workstation is to have a **mksysb** image of the primary control workstation before the **install_hacws** command is executed.

Both the primary and backup control workstations must be running and capable of executing remote commands via the AIX **rsh** command.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error occurred. Diagnostic information is written to standard output and standard error.

Standard output consists of messages indicating the progress of the command as it configures the control workstations.

Security

You must have root privilege to run this command. You must also have SDR administrator and write access to run this command.

Enhanced Security Option

PSSP 3.2 provides the option of running your RS/6000 SP system with an enhanced level of security. This function removes the dependency PSSP has to internally issue **rsh** and **rcp** commands as a root user from a node. When this function is enabled PSSP does not automatically grant authorization for a root user to issue **rsh** and **rcp** commands from a node. If you enable this option some procedures may not work as documented. For example, to run HACMP an administrator must grant the authorizations for a root user to issue **rsh** and **rcp** commands that PSSP would otherwise grant automatically. See the Red Book *Exploiting RS/6000 SP Security: Keeping it Safe* for a description of this function and a complete list of limitations.

Prerequisite Information

Refer to *PSSP: Administration Guide* for information on the HACWS option.

Location

/usr/sbin/hacws/install_hacws

Related Information

PSSP Commands: **install_cw**, **setup_logd**

AIX Commands: **rsh**

Examples

1. To configure both control workstations on an SP system, enter the following:

   ```
   install_hacws -p primary_cw -b backup_cw -s
   ```

2. To configure the control workstations separately, enter the following.

   On the primary control workstation, enter:

   ```
   install_hacws -p primary_cw -b backup_cw
   ```

   After the preceding command completes on the primary control workstation, enter the following on the backup control workstation:

   ```
   install_hacws -p primary_cw -b backup_cw
   ```

kadmin

Purpose

kadmin – Provides network access to Kerberos Version 4 authentication database administration functions.

Syntax

kadmin [−**u** *admin_name*] [−**r** *default_realm*] [−**m**]

Flags

−**u** Specifies a Kerberos Version 4 principal name to use instead of your AIX login name. This *admin_name* must be a valid AIX login name.

−**r** Specify if you want a realm other than the local realm to be the default.

−**m** Allows multiple requests without Kerberos Version 4 reauthentication (reentry of your administrative password).

Operands

None.

Description

This command provides an interactive interface to the primary Kerberos Version 4 authentication database. Administrators use **kadmin** to add new users and services to the database, and to change information about existing database entries. For example, an administrator can use **kadmin** to change a user's password. An administrator is a user with an **admin** instance whose name appears in at least one of the authentication administration Access Control Lists (ACLs).

The **kadmin** program communicates over the network with the **kadmind** program, which runs on the machine housing the primary authentication database. The **kadmind** program creates new entries and makes modifications to the database.

When you enter the **kadmin** command, the program displays a message that welcomes you and explains how to ask for help. Then **kadmin** waits for you to enter commands. After you enter a command, you are prompted to enter your admin password. If the −**m** option is used, you are prompted for your admin password only for the first command entered. You do not need to issue the **k4init** command prior to running this command because the necessary tickets are obtained automatically.

When using the **kadmin** command, the principal's expiration date and maximum ticket lifetime are set to the default values. To override the defaults, the root user must run the **kdb_edit** command to modify those attributes.

Use the **add_new_key** (or **ank** for short) command to add a new principal to the authentication database. The command requires the principal identifier as an argument. The identifier given can be fully qualified using the standard *name.instance@realm* convention. You are asked to enter your admin password and are then prompted twice to enter the principal's new password. If a realm is not specified, the local realm is used unless another was given on the command line with the **r** flag. If no instance is specified, a null instance is used. If a realm other than the default realm is specified, you need to supply your admin password for the specified realm.

Use **change_password** to change a principal's password. The command requires the principal identifier as an argument. You are asked to enter your admin password and are then prompted twice to enter the principal's new password. The identifier given can be fully qualified using the standard *name.instance@realm* convention.

Use the **change_admin_password** to change your admin instance password. This command requires no arguments. It prompts you for your old admin password, then prompts you twice to enter the new admin password. If this is your first command, the default realm is used. Otherwise, the realm used in the last command is used.

Use **destroy_tickets** to destroy any admin tickets obtained by the **kadmin** command.

Use **list_requests** to get a list of possible commands.

Use **help** to display various **kadmin** help messages. If entered without an argument, **help** displays a general help message. You can get detailed information on specific **kadmin** commands by entering **help** *command_name*.

To quit the program, type **quit**.

To invoke a shell command, precede the command with an exclamation point.

Files

/var/kerberos/database/admin_acl.{add,get,mod}
Access Control List files.

Location

/usr/kerberos/bin/kadmin

Related Information

Commands: **add_principal**, **kadmind**, **kpasswd**, **ksrvutil**

Refer to the "RS/6000 SP Files and Other Technical Information" section of *PSSP: Command and Technical Reference* for additional **Kerberos** information.

Examples

The following contains an example of adding a user. To add a user, enter:

kadmin

```
Welcome to the Kerberos Administration Program, version 4

Type "help" if you need it.

admin:  help

Welcome to the Kerberos administration program.Type "?" to get

a list of requests that are available. You can get help on each of

the commands by typing "help command_name". Some functions of this

program requires an "admin" password from you. This is a password

private to you, that is used to authenticate requests from this

program. You can change this password with the "change_admin_password"

(or short form "cap") command. Good Luck!

admin:  ?


Available admin requests:


change_password, cpw        Change a user's password

change_admin_password, cap  Change your admin password

add_new_key, ank            Add new user to kerberos database

get_entry, get              Get entry from kerberos database

destroy_tickets, dest       Destroy admin tickets

help                        Request help with this program

list_requests, lr, ?        List available requests.

quit, exit, q               Exit program.
```

```
admin:  ank mroz

Admin password:

Password for mroz:

Verifying, please re-enter Password for mroz:

mroz added to database.

admin:  q

Cleaning up and exiting.
```

Note: Passwords are not echoed back to the user.

kadmind Daemon

Purpose

kadmind – Contains the daemon for Kerberos Version 4 authentication database administration.

Syntax

kadmind [−**h**] [−**n**] [−**r** *realm*] [−**d** *db_name*] [−**f** *file_name*] [−**a** *acldir*]

Flags

−**h** Specifies that the **kadmind** command list the available subcommands and exit.

−**n** Specifies that the master key from the master key cache file be obtained. Otherwise, it prompts the user to enter the master key interactively.

−**r** Specifies that the **kadmind** command is to service a realm other than the local realm. *realm* is the authentication realm name.

−**d** Specifies an authentication database name other than the default. *db_name* is a directory path.

−**f** Specifies the log file in which the daemon records status and error messages.

−**a** Specifies a directory other than the default that contains the Access Control Lists. *acldir* is a directory path.

Note: Use of the −**r**, −**d**, and −**a** flags with values other than the system defaults is not supported on the SP system.

Operands

None.

Description

The **kadmind** daemon is the Kerberos Version 4 authentication database server for the password-changing and administration tools. It uses the master key for authorization.

The **kadmind** daemon listens for requests on the **kerberos_master/tcp** port. If this port is not defined in the **/etc/services** file, it uses port 751.

When performing requests on behalf of clients, **kadmind** checks access control lists (ACLs) to determine the authorization of the client to perform the requested action. Currently three distinct access types are supported:

- Addition (.add ACL file). A principal on this list can add new principals to the database.
- Retrieval (.get ACL file). A principal on this list can retrieve database entries.

  Note: A principal's private key is never returned by the get functions.
- Modification (.mod ACL file). A principal on this list can modify entries in the database.

Principals are always granted authorization to change their own password.

Files

/.k Master key cache file.

/var/kerberos/database/admin_acl.{add,get,mod}
Access Control List files.

/var/kerberos/database/principal.pag, /kerberos/database/principal.dir
Default files containing the authentication database.

/var/adm/SPlogs/kerberos/admin_server.syslog
Default log file.

Location

/usr/lpp/ssp/kerberos/etc/kadmind

Related Information

Commands: **add_principal**, **kadmin**, **kpasswd**

Refer to the "RS/6000 SP Files and Other Technical Information" section of *PSSP: Command and Technical Reference* for additional **Kerberos** information.

Examples

To see how **kadmind** is started, display the subsystem defintion by issuing:

```
odmget -q subsysname=kadmind SRCsubsys
```

Output will resemble:

```
SRCsubsys:
        subsysname = "kadmind"
        synonym = ""
        cmdargs = "-n"
        path = "/usr/lpp/ssp/kerberos/etc/kadmind"
        uid = 0
        auditid = 0
        standin = "/dev/null"
        standout = "/dev/console"
        standerr = "/dev/console"
        action = 1
        multi = 0
        contact = 2
        svrkey = 0
        svrmtype = 0
        priority = 20
        signorm = 15
        sigforce = 15
        display = 1
        waittime = 20
        grpname = ""
```

kdb_destroy

Purpose

kdb_destroy – Destroys the Kerberos Version 4 authentication database.

Syntax

kdb_destroy

Flags

None.

Operands

None.

Description

The **kdb_destroy** command removes the Kerberos Version 4 authentication database.

You first must reply **y** or **Y** to a prompt to confirm the request, or **kdb_destroy** exits without removing the database files.

This command can only be issued on the system on which the authentication database resides.

Note: This command does not remove database backup files created by the **kdb_util** command nor the **/.k** file created by the **kstash** command.

Files

/var/kerberos/database/principal.pag, /usr/kerberos/database/principal.dir
Files containing the Kerberos Version 4 authentication database.

Security

You must have root privilege to run this command.

Location

/usr/lpp/ssp/kerberos/etc/kdb_destroy

Related Information

Command: **kdb_init**

Examples

To destroy the Kerberos Version 4 authentication database, enter:

```
kdb_destroy
```

kdb_edit

Purpose

kdb_edit – Edits the Kerberos Version 4 authentication database.

Syntax

kdb_edit [–**n**]

Flags

–**n** Specifies that the master key is obtained from the master key cache file. Otherwise, **kdb_edit** prompts the user to enter the master key interactively.

Operands

None.

Description

The **kdb_edit** command is used to create or change principals in the Kerberos Version 4 authentication database. It uses the master key for authorization.

After the master key is verified, **kdb_edit** begins a prompt loop. The user is prompted for the principal name and instance to be modified. If the entry is not found, the user can create it. After an entry is found or created, the user can set the password, expiration date, maximum ticket lifetime, and attributes. Default expiration dates, maximum ticket lifetimes, and attributes are presented in brackets. If the user presses return, the default is selected. There is no default password. The password RANDOM is interpreted specially, and if entered, the program selects a random key for the principal.

You should use random key generation only if you use the **kdb_edit** command to replace a deleted service principal (for example, **rcmd**.*host_name*).

If you enter a ticket lifetime value, it must be a number between 0 and 255. The actual maximum lifetime value that you choose will be between five minutes and 30 days. Refer to *PSSP: Administration Guide* for a complete list of the possible Kerberos Version 4 ticket lifetime values you can enter and the corresponding durations in days, hours, and minutes. The following list shows a representative sample with approximate durations:

```
Response to kdb_edit          Approximate Duration

       141                           1 day

       151                           2 days

       170                           1 week

       180                           2 weeks

       191                           1 month
```

After the entry has been created or changed, “Edit O.K.” is printed.

Files

/.k Master key cache file.

/var/kerberos/database/principal.pag, /usr/kerberos/database/principal.dir
Files containing the Kerberos Version 4 authentication database.

Security

You must have root privilege to run this command.

Location

/usr/lpp/ssp/kerberos/etc/kdb_edit

Related Information

Commands: **kadmin**, **kdb_init**

Refer to the "RS/6000 SP Files and Other Technical Information" section of *PSSP: Command and Technical Reference* for additional **Kerberos** information.

Examples

To add a service from host **mroz**, enter:

kdb_edit -n

```
Opening database...
Previous or default values are in [brackets],
enter return to leave the same, or new value.

Principal name: rcmd
Instance: mroz

<Not found>, Create [y] ? Y

Principal: rcmd, Instance: mroz, kdc_key_ver: 1
New Password:
Verifying, please re-enter
New Password:

Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [1999-12-31] ?
Max ticket lifetime [255] ?
Attributes [0] ?
Edit O.K.
Program re-prompts for another principal "principal name:"

Principal name:

The program exits when no principal name is entered.
```

Note: Passwords are not echoed back to the user.

kdb_init

Purpose

kdb_init – Initializes the Kerberos Version 4 authentication database.

> **Attention**
>
> This command is normally used internally by the **setup_authent** script, and should not be needed otherwise.

Syntax

kdb_init [*realm*]

Flags

None.

Operands

realm Specifies the realm name. If *realm* is not specified, the realm name is set to the local system's network domain name converted to uppercase characters.

Description

Use this command to initialize the Kerberos Version 4 authentication database, creating the necessary initial system principals.

After determining the realm to be created, the command prompts for a master key password. The user should choose a nontrivial, not easily-guessable password. The user **must** remember this password because it is used for other commands. The master key password is used to encrypt every encryption key stored in the database.

Files

/var/kerberos/database/principal.pag, /usr/kerberos/database/principal.dir
Files containing the Kerberos Version 4 authentication database.

Security

You must have root privilege to run this command.

Location

/usr/lpp/ssp/kerberos/etc/kdb_init

Related Information

Commands: **kdb_destroy**, **kdb_edit**, **kdb_util**

Refer to the "RS/6000 SP Files and Other Technical Information" section of *PSSP: Command and Technical Reference* for additional **Kerberos** information.

Examples

To initialize a Kerberos Version 4 database for realm "ABC.ORG", enter:

```
kdb_init ABC.ORG
```

kdb_util

Purpose

kdb_util – The utility program for managing the Kerberos Version 4 authentication database.

Syntax

kdb_util *operation file_name*

Flags

None.

Operands

operation The *operation* must be one of the following:

load Initializes the database with the records described by the text contained in the file *file_name*. Any existing database is overwritten.

dump Dumps the database into a text representation in the file *file_name*.

slave_dump
Performs a database dump similar to the dump operation and creates a semaphore file to indicate to the propagation software that an update is available for distribution to secondary authentication servers.

new_master_key
Prompts for the old and new master key strings, and then dumps the database into a text representation in the file *file_name*. The keys in the text representation are encrypted in the new master key.

file_name Specifies the name of the file.

Description

The **kdb_util** command allows the user to perform various utility operations on the Kerberos Version 4 authentication database.

Files

/var/kerberos/database/principal.pag, /usr/kerberos/database/principal.dir
Files containing the Kerberos Version 4 authentication database.

<**data_file>.ok** Semaphore file created by the **slave_dump** operation.

Security

You must have root privilege to run this command.

Location

/usr/lpp/ssp/kerberos/etc/kdb_util

Related Information

Commands: **kdb_init**, **kprop**, **kpropd**

Refer to the "RS/6000 SP Files and Other Technical Information" section of *PSSP: Command and Technical Reference* for additional **Kerberos** information.

Examples

To dump the contents of the Kerberos Version 4 database in readable ASCII, enter:

```
kdb_util dump /tmp/KerbDump
```

k4destroy, kdestroy

Purpose

k4destroy – Destroys Kerberos Version 4 authentication tickets.

Syntax

k4destroy [–**f**] [–**q**]

Flags

–**f** Indicates that **k4destroy** should not display a status message.

–**q** Indicates that **k4destroy** should display a status message, but should not beep the terminal on an error.

Operands

None.

Description

IBM strongly suggests using the **k4destroy** form. DCE includes the **kdestroy** command, so using **kdestroy** instead of **k4destroy** may produce unexpected results if DCE is installed.

The **k4destroy** command destroys the user's Kerberos Version 4 authentication tickets. The command writes zeros to the user's current ticket cache file and then removes the file from the file system. If the file does not exist or if an error occurs, a message is displayed. The current ticket file is determined by the KRBTKFILE environment variable. If the KRBTKFILE environment variable is undefined, the current ticket file is **/tmp/tkt***uid*, where *uid* specifies your user identification number. If **k4destroy** cannot destroy the ticket file, the command warns you by making your terminal beep. You can place the **k4destroy** command in your **.logout** file (C shell only) so that your tickets are destroyed automatically when you log out.

Files

/tmp/tkt*uid* The default ticket file (*uid* is the decimal UID of the user).

Location

/usr/bin/k4destroy

/usr/lpp/ssp/kerberos/bin/kdestroy

Related Information

Commands: **k4init**, **k4list**

Refer to the "RS/6000 SP Files and Other Technical Information" section of *PSSP: Command and Technical Reference* for additional **Kerberos** information.

Examples

1. This example shows destroying cached Kerberos Version 4 tickets:

```
$k4destroy

Tickets destroyed.

$
```

kerberos Daemon

Purpose

kerberos – The Kerberos Version 4 authentication ticket-granting service daemon.

Syntax

kerberos [–**a** *max_age*] [–**l** *log_file*] [–**m**] [–**n**] [–**p** *pause_seconds*] [–**r** *realm*] [–**s**] [*database*]

Flags

–**a** Specifies the maximum database age. Its value must be between one hour and three days, in seconds. For slave servers, the default is one day. For the primary server, the default is not to check the age of the database.

–**l** Specifies the log file path name. (This is lowercase **l**, as in **l**ist.)

–**m** Prompts for the master key. If the –**m** option is not specified, the master key is obtained from the master key cache file.

–**n** Specifies that the age of the database against maximum not be checked. If desired, this option can override the default for secondary servers.

–**p** Specifies the pause interval. It must be between 5 and 3600 seconds. The default is to hang indefinitely on an error.

–**r** Allows the realm to be specified instead of assuming the local realm.

–**s** Indicates that this server is a secondary (backup) server.

Operands

database Contains the path name of the authentication database.

Note: Specification of a database path name other than the default, **/var/kerberos/database/principal**, is not supported on the SP system.

Description

kerberos is the daemon program that provides the Authentication Service and the Ticket Granting Service to client programs that want to obtain tickets for authenticated services.

The **kerberos** daemon listens for requests on the **kerberos4/upd** port. If this port is not defined in the **/etc/services** file, it uses port 750.

When you start the server (normally from **init**), you can specify a maximum age for the database files. This can be used to ensure that you do not start a secondary server with out-of-date information. This could occur in a situation where a secondary server system was down when a database update was scheduled.

Files

/var/kerberos/database/principal.pag, /var/kerberos/database/principal.dir
Files containing the authentication database.

/.k Master key cache file.

/var/adm/SPlogs/kerberos/kerberos.log, /var/adm/SPlogs/kerberos/kerberos.slave_log
Log files.

Location

/usr/lpp/ssp/kerberos/etc/kerberos

Related Information

Commands: **kdb_init**, **kprop**, **kpropd**

Refer to the "RS/6000 SP Files and Other Technical Information" section of *PSSP: Command and Technical Reference* for additional **Kerberos** information.

Examples

To see how kerberos is started, display the subsystem definition by issuing::

```
odmget -q subsysname=kerberos SRCsubsys
```

Output should resemble:

```
SRCsubsys:
  subsysname = "kerberos"
  synonym = ""
  cmdargs = ""
  path = "/usr/lpp/ssp/kerberos/etc/kerberos"
  uid = 0
  auditid = 0
  standin = "/dev/null"
  standout = "/dev/console"
  standerr = "/dev/console"
  action = 1
  multi = 0
  contact = 2
  svrkey = 0
  svrmtype = 0
  priority = 20
  signorm = 15
  sigforce = 15
  display = 1
  waittime = 20
  grpname = ""
```

kfserver

Purpose

kfserver – Sends keyfiles to requesting node over the s1term.

Syntax

kfserver

Flags

None.

Operands

None.

Description

The **kfserver** script is run from inetd.conf upon request from a node for its Kerberos V4 srvtab file. When the client connects to the server, **kfserver** queries the socket for the node's IP address requesting its srvtab file. **kfserver** locates the srvtab file and sends it over the s1term in write mode.

Files

The log file **/var/adm/SPlogs/kfserver/kfserver.log.***pid* is created.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error or errors occurred. Review the log file.

An unsuccessful run of this command, depending on where an error occurred, will result in the keyfile transfer being unsuccessful.

Security

You must have root privilege to run this command.

Restrictions

This command runs only on the control workstation. This command is not intended to be run manually; it requires a socket opened to it.

This command uses the s1term in write mode. Only one s1term session in write mode is allowed per node, at a time.

Location

/usr/lpp/ssp/install/bin/kfserver

k4init, kinit

Purpose

k4init – Obtains a Kerberos Version 4 authentication ticket.

Syntax

k4init [–**i**] [–**l**] [–**r**] [–**v**] [*name*]

Flags

–i Requests the command to prompt you for an instance, unless one is specified in the *name* operand.

–l Requests the command to prompt you for the ticket lifetime. If not specified, the ticket will have the maximum time allowed for the user. (This is lowercase **l**, as in **l**ist.)

–r Requests the command to prompt you for an authentication realm, unless one is specified in the *name* operand. This option lets you authenticate yourself within an authentication realm other than the local realm.

–v Specifies verbose mode. The name of the ticket file used is printed and a status message indicating whether or not your authentication attempt was successful.

Operands

name Specifies your user principal identifier. The principal name can be qualified with either an instance or a realm *name.instance@realm*, or both. Refer to the **Kerberos** command for details.

Description

IBM strongly suggests using the **k4init** form. DCE includes the **kinit** command, so using **kinit** instead of **k4init** may produce unexpected results if DCE is installed.

The **k4init** command is used to authenticate the user's identify to the SP authentication service. All previous tickets are discarded.

When you use the **k4init** command without options, it prompts for your principal name and password, and tries to authenticate your identity within the local realm. If the specified principal name and password are correct, **k4init** retrieves your initial ticket and puts it in the ticket file specified by your KRBTKFILE environment variable. If the KRBTKFILE variable is undefined, your ticket is stored in the **/tmp/tkt***uid* file, where *uid* specifies your user identification number.

Note: These tickets are shared by all processes running under the user's IDs. The KRBTKFILE environment variable can be set to change the location of the ticket cache file.

If you specify the **–l** flag, the command prompts you to enter a ticket lifetime, in minutes. The actual value you enter will differ somewhat from the actual lifetime, because lifetimes are set to one of a discrete set of values ranging from five minutes to 30 days. **k4init** rounds the value you enter up to the next higher limit, and applies the maximum that is defined for your Kerberos principal. If you enter a value higher than your allowed limit, **k4init** does not indicate an error, but simply assigns your maximum lifetime in the ticket it creates. Refer to *PSSP: Administration Guide* for the complete list of maximum lifetime values that the administrator can set. The following list shows a representative sample of lifetimes you can request:

Response to k4init prompt	Approximate duration
1500	1 day
3000	2 days
10000	1 week
20000	2 weeks
43000	1 month

Depending on your security policy, you may want to use the **k4destroy** command to destroy any active tickets before you end your login session. You can place the **k4destroy** command in your **.logout** file (C shell only) so that your tickets are destroyed automatically when you logout.

The KRBTKFILE environment variable is used to specify the ticket cache file used by **k4init** to store authentication tickets.

Files

/tmp/tkt*uid* The default ticket file (*uid* is the decimal UID of the user).

Location

/usr/bin/k4init

/usr/lpp/ssp/kerberos/bin/kinit

Related Information

Commands: **k4destroy**, **k4list**

Refer to the "RS/6000 SP Files and Other Technical Information" section of *PSSP: Command and Technical Reference* for additional **Kerberos** information.

Examples

1. This example shows logging into Kerberos Version 4 as the principal "richard":

```
$k4init richard

kerberos Initialization for "richard"

Password:

$
```

k4list, klist

Purpose

k4list – Lists currently held Kerberos Version 4 authentication tickets.

Syntax

k4list [–**s** | –**t**] [–**file** *name*] [–**srvtab**]

Parameters

–s Indicates silent mode. The **k4list** command does not print the issue and expire times, the name of the tickets file, or the identity of the principal. This flag is ignored if **srvtab** is specified.

–t Indicates test mode. The **k4list** command just checks for the existence of a nonexpired ticket-granting-ticket. If one is present, it exits with a status of 0. Otherwise, it exits with a status of 1. No output is displayed.

–file Specifies the name of a ticket cache file. When the –**file** option is not specified, the **k4list** command uses the KRBTKFILE environment variable to determine the location of the ticket cache file. If KRBTKFILE is not set, **/tmp/tkt***uid* file is used, where *uid* is the AIX user ID. When **srvtab** is also specified, this flag specifies the name of the server key file whose contents are to be displayed.

–srvtab Specifies that the **k4list** command is to list the contents of a server key file instead of a ticket cache file. If the **file** option is not specified, the default key file is **/etc/krb-srvtab**.

Operands

None.

Description

IBM strongly suggests using the **k4list** form. DCE includes the **klist** command, so using **klist** instead of **k4list** may produce unexpected results if DCE is installed.

The **k4list** command prints the principal name and the name of the file containing the user's tickets. It also lists the principal name, issue time, and expiration time for each service ticket held by the user. Principal names are listed in the form *name.instance@realm*. The period (.) is omitted if the instance is null and the at sign (@) is omitted if the realm is null.

Files

/etc/krb.conf Contains the name of the local realm.

/etc/krb-srvtab The default service key file.

/tmp/tkt*uid* The default ticket file (*uid* is the decimal UID of the user).

Location

/usr/bin/k4list

/usr/lpp/ssp/kerberos/bin/klist

Related Information

Commands: **k4destroy**, **kerberos**, **k4init**

Examples

1. This example shows a listing of the default ticket cache file for the root user (uid 0):

```
# k4list

Ticket file:    /tmp/tkt0

Principal:  root.admin@XYZ.ABC.COM



  Issued           Expires          Principal

Nov 12 16:26:11  Dec 12 16:26:11  krbtgt.XYZ.ABC.COM@XYZ.ABC.COM

Nov 12 16:26:46  Dec 12 16:26:46  hardmon.cwksta@XYZ.ABC.COM

Nov 12 16:45:15  Dec 12 16:45:15  rcmd.cwksta@XYZ.ABC.COM

#
```

   The second line shows the Kerberos principal acting as client, to whom the tickets belong. This is the user principal you supplied to the **k4init** command, or the **rcmd**.*instance* service principal used by **rcmdtgt**. The list of tickets always begins with the ticket-granting-ticket. The others are service tickets; in this case for the System Monitor service on the control workstation (**hardmon**) and the SP Remote Command service also on the control workstation (**rcmd**).

2. This example shows the use of **k4list** to display the key versions for service principals on an SP node:

```
# k4list -srvtab

Server key file:   /etc/krb-srvtab

Service         Instance        Realm      Key Version

------------------------------------------------------

rcmd            node3fi         XYZ.ABC.COM      1

rcmd            node3tr         XYZ.ABC.COM      1

rcmd            node3sw         XYZ.ABC.COM      1

rcmd            node3en         XYZ.ABC.COM      1

#
```

You can determine the versions of service keys in the authentication database by locating the entry for the target service principal in a dump of the SP authentication database. If you have secondary authentication servers, or if you use the procedure for backing up your database that IBM suggests using in *PSSP: Administration Guide*, the database dump can be found in file **/var/kerberos/database/slavesave** on the primary server host.

kpasswd

Purpose

kpasswd – Changes the Kerberos Version 4 principal's password.

Syntax

kpasswd [−**h**] [−**n** *user*] [−**i** *instance*] [−**r** *realm*] [−**u** *full_name*]

Flags

−**h** Specifies that **kpasswd** is to print a brief summary of the options and then exit.

−**n** Specifies the name to be used as the principal name rather than the user name of the user running **kpasswd**. (This is determined from the ticket file if it exists; otherwise, it is determined from the AIX login name.)

−**i** Specifies the instance to be used as the instance of the user principal, rather than a null instance.

−**r** Specifies the realm to be used as the realm rather than the local realm.

−**u** Specifies a fully qualified principal identifier in the form *name.instance@realm*.

Operands

None.

Description

The **kpasswd** command changes a principal's password.

It prompts for the principal's current password. If the old password is correct, the user is prompted twice for a new password. A message is printed indicating whether or not the password changing operation was successful.

Location

/usr/kerberos/bin/kpasswd

Related Information

Commands: **kadmin**, **k4init**, **passwd**

Refer to the "RS/6000 SP Files and Other Technical Information" section of *PSSP: Command and Technical Reference* for additional **Kerberos** information.

kprop

Purpose

kprop – The network utility to propagate the Kerberos Version 4 authentication database to secondary servers.

Syntax

kprop [–**force**] [–**realm** *realm_name*] *data_file hosts_file*

Flags

–force Overrides the timestamp checking, forcing transmittal even if the database was not modified since last sent.

–realm Allows the realm to be specified instead of assuming the local realm.

Operands

data_file Specifies the file containing the dumped image of the Kerberos Version 4 authentication database produced by the **kdb_util slave_dump** command.

hosts_file Contains a list of secondary server hosts that provide backup to this server.

Description

The **kprop** command reads a list of secondary host names and connects to each one in turn using the **kprop** service provided by the **kpropd** program. The *data_file* (the Kerberos Version 4 authentication database) is transferred if it has been modified since it was last sent successfully.

Files

<**data_file>.ok** Semaphore file created by the **kdb_util slave_dump** operation.

Security

You must have root privilege to run this command.

Location

/usr/lpp/ssp/kerberos/etc/kprop

Related Information

Commands: **kdb_util**, **kerberos**, **kpropd**

Examples

To force propogation from the primary server to a backup, enter:

```
rm /var/kerberos/database/slavesave.dump_ok

kdb_util slave_dump /var/kerberos/database/slavesave

kprop /var/kerberos/database/slavesave /var/kerberos/database/slavelist
```

kpropd Daemon

Purpose

kpropd – The daemon that receives updates for a secondary Kerberos Version 4 authentication database.

Syntax

kpropd [–**r** *realm*] [–**s** *srvtab*] [–**l** *log_file*] [–**d** *database_name*] *file_name*

Flags

–**r** Overrides the default local realm.

–**s** Overrides the default srvtab name **/etc/krb-srvtab**.

–**l** Specifies a log file name to be used instead of the default. (This is lowercase **l**, as in **l**ist.)

–**d** Specifies the path name of the database.

Note: Use of the –**r**, –**s**, and –**d** flags with values other than the system defaults is not supported on the SP system.

Operands

file_name
Specifies the name of the file to receive from the transmitting host, and then to input to a **kdb_util load** command.

Description

kpropd runs as a daemon on secondary Kerberos Version 4 authentication database server hosts, listening for a TCP connection on the **krb_prop** service.

The **kpropd** daemon listens for requests on the **krb_prop/tcp** port. If this port is not defined in the **/etc/services** file, it uses port 754. It validates the connection, which must be from an administrative host as defined in the **krb.conf** file for the local realm. The service name used for mutual authentication is **rcmd**.

Files

/etc/krb.conf Contains the name of the local realm.

/etc/krb-srvtab Default server key file.

/var/kerberos/database/principal.pag, /var/kerberos/database/principal.dir
Default location of database files.

/var/adm/SPlogs/kerberos/kpropd.log
Log file.

Location

/usr/lpp/ssp/kerberos/etc/kpropd

Related Information

Commands: **kdb_util**, **kerberos**, **kprop**

Examples

To see how **kpropd** is started, display the subsystem definition by issuing:

```
odmget -q subsysname=kerberos SRCsubsys
```

Ouput should resemble:

```
SRCsubsys:

  subsysname = "kpropd"

  synonym = ""

  cmdargs = "/var/kerberos/database/slavesave"

  path = "/usr/lpp/ssp/kerberos/etc/kpropd"

  uid = 0

  auditid = 0

  standin = "/dev/null"

  standout = "/dev/console"

  standerr = "/dev/console"

  action = 1

  multi = 0

  contact = 2

  svrkey = 0

  svrmtype = 0

  priority = 20

  signorm = 15

  sigforce = 15

  display = 1

  waittime = 20

  grpname = ""
```

ksrvtgt

Purpose

ksrvtgt – Obtains a Kerberos Version 4 authentication ticket with a maximum allowed lifetime.

Syntax

ksrvtgt *name instance* [[*realm*] *srvtab*]

Flags

None.

Operands

name
instance
realm Specifies the principal as *name.instance@realm* (where realm defaults to the local realm defined in **/etc/krb.conf**).

srvtab Specifies the service key file to use (defaults to **/etc/krb-srvtab**).

Description

The **ksrvtgt** command retrieves Kerberos Version 4 authentication with a maximum lifetime, decrypts the response using the service key found in the service key file, and stores the ticket in the ticket cache file.

This command is intended primarily for use in shell scripts and other batch-type facilities.

The KRBTKFILE environment variable is used to specify the ticket cache file used by **ksrvtgt** to store authentication tickets.

If Kerberos Version 4 is not an active authentication method for AIX remote commands, and Compatibility is not an active authentication method for SP trusted services, this command performs no function but returns successfully.

Environment Variables

KRBTKFILE
The pathname of the ticket cache file to use.

Files

/etc/krb.conf
Contains the name of the local realm and the names of the servers.

/etc/krb-srvtab
The default service key file.

Standard Error

Output consists of error messages, when the command cannot complete successfully.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error occurred.

Security

The **ksrvtgt** command can be run by any user who owns a service key file.

Location

/usr/kerberos/bin/ksrvtgt

Related Information

Commands: **ksrvutil**, **k4destroy**, **k4init**

Refer to the "RS/6000 SP Files and Other Technical Information" section of *PSSP: Command and Technical Reference* for additional **Kerberos** information.

ksrvutil

Purpose

ksrvutil – Manipulates a server Kerberos Version 4 key file.

Syntax

ksrvutil [−**afs** | −**krb**] [−**k**] [−**i**] [−**f** *file_name*] *operation*

Flags

−afs Indicates that the Kerberos Version 4 authentication database is being managed by AFS, and that the key file should be constructed to work with AFS.

−krb Indicates that the Kerberos Version 4 authentication database is being managed by a server compatible with the MIT version of Kerberos, and that the key file should be constructed to work with that version.

If neither −**afs** nor −**krb** are specified, the value of the System Data Repository (SDR) **authent_server** attribute is used. If the value of the SDR **authent_server** attribute cannot be obtained, the default is −**krb**.

−k When specified for the list operation, keys are also displayed. For the change operation, the old and new keys are displayed. For the add operation, the key is displayed.

−i Prompts for yes or no before changing each key.

−f For all operations, specifies the server key file to update. The default is **/etc/krb-srvtab**.

Note: Specification of a **srvtab** file other than the system default is not supported on the SP system.

Operands

operation The operation must be one of the following:

list Lists the version number and principal name in the server key file.

change Changes all the keys in the server key file.

add Adds a server principal name and key to the server key file. The command prompts for name, instance, realm, and key version number, and asks for a password. The **ksrvutil** command then converts the password to a key and appends the key file with the new information.

delete Deletes keys in the key file. The user is prompted before deleting each key.

Description

The **ksrvutil** command allows an administrator to list or change keys currently in the key file or to add new keys to the key file.

The **ksrvutil** command always backs up the key file before making any changes. If **ksrvutil** is unsuccessful during a change or add operation, you can recover a usable key file by appending the workfile containing the new and changed keys, *file_name*.work to the backup copy or the original, *file_name*.old, and replacing the key file *file_name* with the result, for example:

```
cat /etc/krb-srvtab.old /etc/krb-srvtab.work >/etc/krb-srvtab
```

The recovered key file can be used, but it may contain some out-of-date keys.

Files

/etc/krb-srvtab Default server key file.

Security

You must have root privilege to run this command.

Location

/usr/kerberos/bin/ksrvutil

Related Information

Commands: **kadmin**, **ksrvtgt**, **rcmdtgt**

kstash

Purpose

kstash – Saves the system's Kerberos Version 4 authentication master key.

Syntax

kstash

Flags

None.

Operands

None.

Description

The **kstash** command saves the system's Kerberos Version 4 authentication database master key in the master key cache file. The user is prompted to enter the master key (the same one as specified to **kdb_init**) to verify the authenticity of the key and authorize caching it.

Files

/.k Master key cache file.

/var/kerberos/database/principal.pag, /var/kerberos/database/principal.dir
Files containing the authentication database.

Security

You must have root privilege to run this command.

Related Information

Command: **kdb_init**

lppdiff

Purpose

lppdiff – Queries installed Licensed Program Products (LPPs) on a group of hosts.

Syntax

lppdiff [−**Gvacn**] [−**l** *login*] [−**w** *collective*] [−**f** *fanout*]
[*fileset* [*fileset* ...] | **all**]

lppdiff [−**h**]

Flags

−G Expands the scope of the −**a** flag to include all nodes in the SP system. The −**G** flag is meaningful only if used in conjunction with the −**a** flag.

−v Verifies hosts before adding to the working collective. If this flag is set, each host to be added to the working collective is checked before being added.

−a Specifies that the System Data Repository (SDR) **initial_hostname** field for all nodes in the current system partition be added to the working collective. If −**G** is also specified, all nodes in the SP system are included.

−c Displays information as a list separated by colons. Note: Error messages displayed may contain a colon.

−n Displays the count of the number of nodes with a fileset in a given state. (This is the default.)

−l Specifies a remote user name under which to execute the query. If −**l** is not used, the remote user name is the same as your local user name.

−w Specifies a list of host names, separated by commas, to include in the working collective. Both this flag and the −**a** flag can be included on the same command line. Duplicate host names are included only once in the working collective.

−f Specifies a fanout value. The default value is 64. This indicates the maximum number on concurrent rsh's to execute. Sequential execution can be specified by indicating a fanout value of 1. The fanout value is taken from the FANOUT environment variable if the −**f** flag is not specified, otherwise the default is used.

−h Displays usage information.

Operands

fileset Specifies the LPP to query. Using **all** for this operand will query all LPPs installed on the host.

Description

Use this command to query the status of installed LPPs on a group of hosts. The output from each host is collected and identical results are compressed to show the names and a count of the hosts that had identical results.

The **dsh** command is used to execute the queries on the remote hosts. The **lslpp** command is used to get the status of the installed LPPs on the remote hosts. The **lslpp** command is called on each host with the −**l**, −**a**, −**c**, and −**q** flags.

Output from the **lppdiff** command consists of one entry for each unique LPP listing information about that LPP. Each LPP's entry is followed by a list of all hosts that have that LPP installed. An LPP is considered unique if any one of the components in its description differ from that of another. For example, consider two hosts that both have ssp.basic installed. On host 1, it is in the APPLY state and on host 2, it is in the COMMITTED state. These LPPs are considered unique and, therefore, each will get its own set of output from **lppdiff**.

The flags for **lppdiff** are used to direct the **dsh** command to certain hosts and to control its behavior. See the **dsh** command for details on these flags and how to use them.

The *fileset* operand to **lppdiff** can be one of two things. It can either be **all** which queries and displays information about all LPPs installed on the specified hosts, or it can be the name of a file set to query on the specified hosts. The "*" character can be used to specify multiple file sets. For example, **lppdiff** −**Ga ssp.*** queries any file sets starting with "ssp." on all hosts in the system.

| ## Security

| You must have access to the AIX Secure Remote Commands to run this command.

Location

/usr/lpp/ssp/bin/lppdiff

Examples

1. To query LPP information for ssp.basic on all nodes in the current system partition, enter:

   ```
   [k22s] > lppdiff -a ssp.basic
   ```

   You should receive output similar to the following:

```
-------------------------------------------------------------------
      Name       Path               Level    PTF   State    Type Num
-------------------------------------------------------------------
LPP:  ssp.basic  /etc/objrepos      2.3.0.0        COMMITTED  I   5
From: k22n09.ppd.pok.ibm.com k22n10.ppd.pok.ibm.com k22n11.ppd.pok.
         ibm.com k22n12.ppd.pok.ibm.com k22n13.ppd.pok.ibm.com
-------------------------------------------------------------------
LPP:  ssp.basic  /usr/lib/objrepos  2.3.0.0        COMMITTED  I   5
From: k22n09.ppd.pok.ibm.com k22n10.ppd.pok.ibm.com k22n11.ppd.pok.
         ibm.com k22n12.ppd.pok.ibm.com k22n13.ppd.pok.ibm.com
```

2. To query LPP information for all options starting with X11.base on a specific node, enter:

```
[k22s] > lppdiff -w k22n01 X11.base*

-------------------------------------------------------------------

      Name         Path             Level   PTF  State    Type Num

-------------------------------------------------------------------

LPP:  X11.base.rte /etc/objrepos    4.1.4.0      COMMITTED  I   1

From: k22n01

-------------------------------------------------------------------

LPP:  X11.base.smt /etc/objrepos    4.1.4.0      COMMITTED  I   1

From: k22n01

-------------------------------------------------------------------

LPP:  X11.base.   /usr/lib/objrepos 4.1.0.0      COMMITTED  I   1

      common

From: k22n01

-------------------------------------------------------------------

LPP:  X11.base.lib /usr/lib/objrepos 4.1.4.0     COMMITTED  I   1

From: k22n01

-------------------------------------------------------------------

LPP:  X11.base.rte /usr/lib/objrepos 4.1.4.0     COMMITTED  I   1

From: k22n01

-------------------------------------------------------------------

LPP:  X11.base.smt /usr/lib/objrepos 4.1.4.0     COMMITTED  I   1

From: k22n01
```

3. To query LPP information for ssp.clients and ssp.bogus (a nonexistent file set) on all nodes in the system, enter:

```
[k22s] > lppdiff -Ga ssp.clients ssp.bogus

-------------------------------------------------------------------

      Name        Path              Level    PTF State    Type Num

-------------------------------------------------------------------

LPP:  ssp.clients  /etc/objrepos     3.1.0.0     COMMITTED  I    4

From: k22n03.ppd.pok.ibm.com k22n04.ppd.pok.ibm.com k22n07.ppd.pok.

         ibm.com k22n08.ppd.pok.ibm.com

-------------------------------------------------------------------

LPP:  ssp.clients  /etc/objrepos     2.4.0.6     APPLIED    F    4

From: k22n03.ppd.pok.ibm.com k22n04.ppd.pok.ibm.com k22n07.ppd.pok.

         ibm.com k22n08.ppd.pok.ibm.com

-------------------------------------------------------------------

LPP:  ssp.clients  /etc/objrepos     2.3.0.0     COMMITTED  I    8

From: k22n01.ppd.pok.ibm.com k22n05.ppd.pok.ibm.com k22n06.ppd.pok.

         ibm.com k22n09.ppd.pok.ibm.com k22n10.ppd.pok.ibm.com

         k22n11.ppd.pok.ibm.com k22n12.ppd.pok.ibm.com k22n13.ppd.

         pok.ibm.com

-------------------------------------------------------------------

LPP:  ssp.clients  /usr/lib/objrepos 3.1.0.0     COMMITTED  I    4

From: k22n03.ppd.pok.ibm.com k22n04.ppd.pok.ibm.com k22n07.ppd.pok.

         ibm.com k22n08.ppd.pok.ibm.com

-------------------------------------------------------------------

LPP:  ssp.clients  /usr/lib/objrepos 2.4.0.6     APPLIED    F    4

From: k22n03.ppd.pok.ibm.com k22n04.ppd.pok.ibm.com k22n07.ppd.pok.

         ibm.com k22n08.ppd.pok.ibm.com
```

```
-------------------------------------------------------------------
LPP:  ssp.clients  /usr/lib/objrepos 2.3.0.0     COMMITTED  I    8
From: k22n01.ppd.pok.ibm.com k22n05.ppd.pok.ibm.com k22n06.ppd.pok.
         ibm.com k22n09.ppd.pok.ibm.com k22n10.ppd.pok.ibm.com
         k22n11.ppd.pok.ibm.com k22n12.ppd.pok.ibm.com k22n13.ppd.
         pok.ibm.com

========================= Errors ===================================
Error: /bin/lslpp: Fileset ssp.bogus not installed.
From:  k22n01.ppd.pok.ibm.com k22n05.ppd.pok.ibm.com k22n06.ppd.pok.
          ibm.com k22n03.ppd.pok.ibm.com k22n04.ppd.pok.ibm.com
          k22n07.ppd.pok.ibm.com k22n08.ppd.pok.ibm.com k22n09.ppd.
          pok.ibm.com k22n10.ppd.pok.ibm.com k22n11.ppd.pok.ibm.com
          k22n12.ppd.pok.ibm.com k22n13.ppd.pok.ibm.com
-------------------------------------------------------------------
```

4. To query LPP information for ssp.clients and ssp.bogus (a non-existent file set) on all nodes in the system, and have the results displayed as a list separated by colons, enter:

```
[k22s] > lppdiff -Gac ssp.clients ssp.bogus


-------------------------------------------------------------------

      Name         Path          Level      PTF State    Type Num

-------------------------------------------------------------------

      k22n03.ppd.pok.ibm.com,k22n04.ppd.pok.ibm.com,

           k22n07.ppd.pok.ibm.com,k22n08.ppd.pok.ibm.com:ssp.clients:

        /etc/objrepos           3.1.0.0    COMMITTED     I     4

-------------------------------------------------------------------

      k22n03.ppd.pok.ibm.com,k22n04.ppd.pok.ibm.com,

        k22n07.ppd.pok.ibm.com,k22n08.ppd.pok.ibm.com:ssp.clients:

        /etc/objrepos           2.4.0.6    APPLIED       F     4

-------------------------------------------------------------------

      k22n01.ppd.pok.ibm.com k22n05.ppd.pok.ibm.com

        k22n06.ppd.pok.ibm.com,k22n09.ppd.pok.ibm.com

        k22n10.ppd.pok.ibm.com k22n11.ppd.pok.ibm.com,

        k22n12.ppd.pok.ibm.com k22n13.ppd.pok.ibm.com:ssp.clients:

        /etc/objrepos           2.3.0.0    COMMITTED     I     8

-------------------------------------------------------------------

      k22n03.ppd.pok.ibm.com k22n04.ppd.pok.ibm.com

        k22n07.ppd.pok.ibm.com,k22n08.ppd.pok.ibm.com:ssp.clients:

        /usr/lib/objrepos       3.1.0.0    COMMITTED     I     4

-------------------------------------------------------------------

      k22n03.ppd.pok.ibm.com k22n04.ppd.pok.ibm.com

        k22n07.ppd.pok.ibm.com,k22n08.ppd.pok.ibm.com:ssp.clients:

        /usr/lib/objrepos       2.4.0.6    APPLIED       F     4
```

```
-------------------------------------------------------------------
      k22n01.ppd.pok.ibm.com k22n05.ppd.pok.ibm.com
        k22n06.ppd.pok.ibm.com,k22n09.ppd.pok.ibm.com
        k22n10.ppd.pok.ibm.com k22n11.ppd.pok.ibm.com,
        k22n12.ppd.pok.ibm.com k22n13.ppd.pok.ibm.com:ssp.clients:
        /usr/lib/objrepos       2.3.0.0    COMMITTED     I     8
========================= Errors ===================================
Error: /bin/lslpp: Fileset ssp.bogus not installed.
From:  k22n01.ppd.pok.ibm.com k22n05.ppd.pok.ibm.com
        k22n06.ppd.pok.ibm.com,k22n03.ppd.pok.ibm.com
        k22n04.ppd.pok.ibm.com k22n07.ppd.pok.ibm.com,
        k22n08.ppd.pok.ibm.com k22n09.ppd.pok.ibm.com
        k22n10.ppd.pok.ibm.com,k22n11.ppd.pok.ibm.com
        k22n12.ppd.pok.ibm.com k22n13.ppd.pok.ibm.com:
```

lsauthpar

Purpose

lsauthpar – Lists and verifies the active remote command authentication methods for a system partition.

Syntax

lsauthpar [–**h** | [–**c**] [–**p** *partition*] [–**v**]]

Flags

–h
Specifies that the command syntax is to be listed. When this flag is specified, other flags are ignored.

–c
Specifies that the output is to be a single line consisting of the colon delimited list of methods, as stored in raw form in the System Data Repository.

–p *partition*
Specifies the partition for which the active authentication methods are to be listed. The partition can be specified in either hostname or IP address format. If none is specified, the value of the **SP_NAME** environment variable will be used. If this option is not specified and **SP_NAME** is not set, the default partition is assumed on the control workstation and the primary partition is assumed on a node.

–v
Specifies that the command should verify the active methods by running **lsauthent** on each accessible node in the partition against the settings stored in the System Data Repository. The –**c** flag is incompatible with this flag.

Operands

None.

Description

The **lsauthpar** command lists the remote command authentication methods that are active for the system partition. This command verifies the settings in the System Data Repository, reporting spurious settings; to make corrections use the **chauthpar** command.

This command also provides an option to check whether all nodes in the applicable system partition have the correct setting. No remote verification will occur, unless the SDR setting is valid.

Because the verification of settings on running nodes is performed using the AIX **rsh** command to execute **lsauthent**, the control workstation and the nodes must have at least one common remote command authentication method active. When this is not the case, verification will result in errors reported by **dsh** or **rsh**. The AIX setting on these nodes must then be reset by the local root user issuing **chauthent** on each node, or by a re–boot. No attempt is made to verify nodes that are inaccessible.

When you specify –**v**, this command uses the remote command facilities to obtain the settings from nodes in the system partition. Authorization to perform that task will be unsuccessful for any remote host that does not have at least one of the required remote command authentication methods active. If the command finds discrepancies, it will be necessary to execute the **chauthent** command on the indicated nodes or to re–boot the nodes, to enable client/server communication with them.

Environment Variables

The **SP_NAME** variable can be used to designate the applicable partition.

Standard Output

When –**v** is omitted, output consists of the methods that are active for the system partition, in the order of precedence established when they were set. When the –**c** flag is specified, the output is a single line consisting of a colon–separated list of methods. Without the –**c** flag, each method is identified by a descriptive name on a separate line. When verification is specified, methods are displayed in colon–delimited format.

Standard Error

Output consists of error messages when the command cannot complete successfully and when discrepancies are found during verification.

Exit Values

0 Indicates successful completion of the command.

1 Indicates that an error occurred.

Security

When the –**v** option is omitted, the **lsauthpar** command may be used by any user. Successful verification of the setting on a node requires the user to be authorized for remote command execution from the control workstation to that node.

Restrictions

When the –**v** option is specified, the command may be executed only on the control workstation.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP) (fileset **ssp.clients**).

Prerequisite Information

The chapters on security in the *PSSP: Administration Guide*.

Location

/usr/lpp/ssp/bin/lsauthpar

Examples

1. To list the remote command authentication methods for the current partition, enter:

```
$lsauthpar

Kerberos 5

Kerberos 4

Standard Aix
```

2. To list in colon format the remote command authentication methods for a partition known by IP address, enter:

```
$lsauthpar -c -p 120.14.89.10

k4:std
```

3. This example verifies authentication methods for a named partition, with no discrepancies found:

```
$lsauthpar –v –p sp4partA

Remote command authentication methods for the partition: k4:std

No discrepancies were found.
```

4. This example verifies the authentication methods for a named partition, where a node's settings do not match the information found in the System Data Repository:

```
$lsauthpar –p sp3p2 –v

Remote command authentication methods for the partition: k5:k4:std

lsauthpar: 0016–347 On sp3n13.abc.com the remote command authentication

methods are incorrectly set to "std"
```

lsauthpts

Purpose

lsauthpts – Lists and verifies the active trusted services authentication methods for a system partition.

Syntax

lsauthpts [−**h** | [−**c**] [−**p** *partition*] [−**v**]]

Flags

−**h** Specifies that the command syntax is to be listed. When this flag is specified, other flags are ignored.

−**c** Specifies single line output consisting of the colon delimited list of methods, as stored in raw form in the System Data Repository.

−**p** *partition*
Specifies the partition for which the active authentication methods are to be listed. The partition can be specified in either hostname or IP address format. If none is specified, the value of the SP_NAME environment variable will be used. If this option is not specified and SP_NAME is not set, the default partition is assumed on the control workstation and the primary partition is assumed on a node.

−**v** Specifies that the command should verify the active methods by running **lsauths** on each accessible node in the partition, against the settings stored in the System Data Repository. The −**c** flag is incompatible with this flag.

Operands

None.

Description

The **lsauthpts** command lists the trusted services authentication methods that are active for the system partition. This command always verifies the correctness of the setting in the System Data Repository, reporting spurious settings to the user, who can use the **chauthpts** command to make corrections.

This command also provides an option to check whether all nodes in the applicable system partition have the correct setting. No remote verification will occur, unless the SDR setting is valid.

Because the verification of settings on running nodes is performed using the AIX **rsh** command to execute **lsauths**, the control workstation and the nodes must have at least one common remote command authentication method active. When this is not the case, verification will result in errors reported by **dsh** or **rsh**. The AIX setting on these nodes must then be reset by the local root user issuing **chauths** on each node, or by a reboot.

Environment Variables

The **SP_NAME** variable can be used to designate the applicable partition.

Standard Output

When −**v** is omitted, output consists of the methods that are active for the system partition, in the order of precedence established when they were set. When the −**c** flag is specified, the output is a single line consisting of a colon−separated list of methods. Without the −**c** flag, each method is identified by a descriptive name on a separate line. When verification is specified, methods are displayed in colon−delimited format.

Standard Error

Output consists of error messages when the command cannot complete successfully, and when discrepancies are found during verification.

When you specify −**v**, this command uses the remote command facilities to obtain the settings from nodes in the system partition. Authorization to perform that task will fail for any remote host that does not have at least one of the required remote command authentication methods active. If the command finds discrepancies, it will be necessary to execute the **chauthts** command on the indicated nodes, or to re−boot the nodes to enable client-server communication.

Exit Values

0 Indicates successful completion of the command.

1 Indicates that an error occurred.

Security

When −**v** is omitted, the **lsauthpts** command may be used by any user. Successful verification of the setting on a node requires the user to be authorized for remote command execution from the control workstation to that node.

Restrictions

When the −**v** option is specified, the command may be executed only on the control workstation.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP) (fileset **ssp.clients**).

Prerequisite Information

The chapters on security in the *PSSP: Administration Guide*.

Location

/usr/lpp/ssp/bin/lsauthpts

Examples

1. Listing the trusted services authentication methods for the current partition:

```
$lsauthpts

Compatibility
```

2. Listing in colon format the trusted services authentication methods for a partition known by IP address:

```
$lsauthpts -c -p 120.14.89.10

dce:compat
```

3. Verifying authentication methods for the default or current partition, with no discrepancies found:

```
$lsauthpts -v

Trusted services authentication methods for the partition: dce

No discrepancies were found.
```

4. Verifying the authentication methods for a named partition, where a node's settings do not match the information found in the System Data Repository:

```
$lsauthpts -p sp3p2 -v

Trusted services authentication methods for the partition: dce:compat

lsauthpts: 0016-347 On sp3n13.abc.com the trusted services authentication

methods are incorrectly set to "compat"
```

lsauthts

Purpose

lsauthts – Lists the active authentication methods for trusted services on a host.

Syntax

lsauthts [–**h** | –**c**]

Flags

–**h** Specifies that the command syntax is to be listed. When this flag is specified, all other flags are ignored.

–**c** Specifies that the output is to be displayed in colon (short) format.

Operands

None.

Description

The **lsauthts** command lists the authentication methods that are used by trusted services on the local host. Trusted services that support multiple methods will attempt to authenticate and authorize client requests using the methods in the order listed

Files

If the file containing the local settings contains invalid data, it will be removed.

Standard Input

The local settings are obtained from the **/spdata/sys1/spsec/auth_methods** file.

Standard Output

The long form of the output consists of a line for each authentication method that is active. The short form consists of one line with the methods separated by colons.

Standard Error

Output consists of error messages, when the command cannot complete successfully.

There are no unique consequences of command errors.

Exit Values

0 Indicates successful completion of the command.

1 Indicates the file does not exist, cannot be accessed, or contains invalid data.

Security

The **lsauthts** command may be used by any user.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP) (fileset **ssp.clients**).

Prerequisite Information

The chapters on security in the *PSSP: Administration Guide*.

Location

/usr/lpp/ssp/bin/lsauthts

Examples

1. Listing the authentication methods on a host that uses just DCE for trusted services authentication:

   ```
   $lsauthts

   DCE
   ```

2. Listing the authentication methods on a host that uses DCE and Compatibility methods for trusted services authentication:

   ```
   $lsauthts -c

   dce:compat
   ```

lsfencevsd

Purpose

lsfencevsd – Lists IBM Virtual Shared Disks that are fenced from access by nodes.

Syntax

lsfencevsd

Flags

None.

Operands

None.

Description

Use this command to display a map that shows which IBM Virtual Shared Disks are fenced from which nodes in the system or system partition.

Security

You must be in the AIX **bin** group to run this command.

Prerequisite Information

PSSP: Managing Shared Disks

Location

/usr/lpp/csd/bin/lsfencevsd

Related Information

Commands: **fencevsd**, **unfencevsd**

Examples

To display the map of fenced IBM Virtual Shared Disks in the system, enter:

```
lsfencevsd
```

The system displays a map similar to the following:

```
minor        Fenced Nodes

(13):           13 14

(14):            1  2
```

lshacws

Purpose

lshacws – Gets the HACWS state of the control workstation.

Syntax

lshacws

Flags

None.

Operands

None.

Description

Use this command to print the current HACWS state of the control workstation. It prints to standard output a number string that indicates the HACWS state of the local host and whether the local host is part of an HACWS configuration.

This command is valid only when issued on the control workstation. When the command is executed and the calling process is not on a control workstation, an error occurs.

Note: The High Availability Cluster Multiprocessing (HACMP) event scripts and installation scripts supplied with the High Availability Control Workstation (HACWS) option of the IBM Parallel System Support Programs for AIX (PSSP), set the control workstation state. The state is changed during reintegration in the HACWS supplied pre- and post-event scripts for HACMP. The administrator should not normally have to set the control workstation state.

Exit Values

0 Indicates successful completion of the command.

1 Indicates that the command could not obtain the HACWS state.

2 Indicates that the command retrieved an HACWS state that was not valid.

3 Indicates that the command was not executed on a control workstation.

The following are the valid printed values and their defined HACWS state:

0 Indicates that this is a control workstation that is not part of an HACWS configuration.

1 Indicates that this is the primary control workstation, but not the active control workstation.

2 Indicates that this is the primary and active control workstation.

16 Indicates that this is the backup control workstation and not the active control workstation.

32 Indicates that this is the backup and active control workstation.

Prerequisite Information

Refer to *PSSP: Administration Guide* for information on the HACWS option.

Location

/usr/lpp/ssp/bin/lshacws

Related Information

Command: **sethacws**

Subroutines: **hacws_set**, **hacws_stat**

Examples

1. The following **lshacws** command results indicate the node type:

```
lshacws

Results: 32 - node is a backup and active control workstation

Results: 16 - node is a backup and inactive control workstation

Results: 2  - node is a primary and active control workstation

Results: 1  - node is a primary and inactive control workstation

Results: 0  - node is a control workstation but not an HACWS configuration

Results: error occurs with exit value = 3  - node is not a control workstation
```

lshsd

Purpose

lshsd – Displays configured hashed shared disks for a virtual shared disk and the characteristics.

Syntax

lshsd [–**l** | –**s**] [*hsd_name* ...]

Flags

–**l** Lists the minor number, the stripe size, the number of virtual shared disks, the name of the hashed shared disk, and the underlying virtual shared disks. (This is lowercase **l**, as in **l**ist.)

–**s** Displays the statistics of reads and writes on underlying virtual shared disks in hashed shared disks.

Operands

hsd_name **...** Specifies a hashed shared disk for the virtual shared disk.

Description

This command displays the configured hashed shared disks. If a list of hashed shared disks follow the flag then information about them is displayed. **lshsd** without any arguments or flag lists the names of all the hashed shared disks currently configured.

Prerequisite Information

PSSP: Managing Shared Disks

Location

/usr/lpp/csd/bin/lshsd

Related Information

Commands: **cfghsd**, **ucfghsd**, **hsdatalst**, **updatehsd**

Examples

1. To list all the configured hashed shared disks, enter:

   ```
   lshsd
   ```

   The system displays a message similar to the following:

   ```
   hsd1

   hsd2

   hsd3

   ⋮
   ```

2. To list hashed shared disks and their characteristics, enter:

   ```
   lshsd -l hsd1 hsd2
   ```

   The system displays a message similar to the following:

   ```
   HSD_name=hsd1 Stripe_size=32768 Hsd_minorno=1 numVsds=2

        option=protectlvcb size_in_MB=40

   vsd.rlv01

   vsd.rlv02



   HSD_name=hsd2 Stripe_size=32768 Hsd_minorno=1 numVsds=3

        option=protectlvcb size_in_MB=40

   vsd.rlv03

   vsd.rlv04

   vsd.rlv05
   ```

3. To list statistical information about hashed shared disk hsd1, enter:

   ```
   lshsd -s hsd1
   ```

   The system displays a message similar to the following:

   ```
               9     hsd parallelism

               0     READ requests not at page boundary

               0     WRITE requests not at page boundary

   HSD_name=hsd1 Stripe_size=4096 HSD_minorno=1 numVSDs=2

   option=protect_lvcb size_in_MB=40

   number_read number_write vsd_name

       16          16       vsdn01v1

       16          16       vsdn02v1
   ```

lskp

Purpose

lskp – Lists Kerberos principals.

Syntax

lskp [–**h** | –**p** | –**s** | –**c** | {*name*[*.instance*]|*name.*|*.instance*} ...]

Flags

–**h** Displays usage information.

–**p** Lists the four principals that are predefined by Kerberos.

–**s** Lists service principals for the **rcmd** and **hardmon** services.

–**c** Lists client principals (all but those listed by –**p** and –**s**).

Operands

{*name***[***.instance***]|** *name.***|***.instance***} ...**
Identifies specific principals to list. Specify *name.* to list all principals with a specific principal name or *.instance* to list all principals with a specific instance.

Note: The *name* must be followed by a period and the *instance* must be preceded by a period.

This operand and the various flags are mutually exclusive. When the command is issued with no operands or flags, all principals are listed.

Description

Use this command to list principals in the local Kerberos database, displaying for each the principal name and instance, the maximum ticket lifetime, and the expiration date. You can list the entire authentication database, an individual entry, all entries with a specified principal name, or all entries with a specified instance. Or you can list entries in different categories: all client (user) principals, all service principals, or all principals predefined by Kerberos.

Files

/var/kerberos/database/admin_acl.get
Access control list for **kadmin** and **lskp**..

/var/kerberos/database/principal.*
Kerberos database files.

Standard Output

For each principal, the **lskp** command displays the principal identifier as *name.instance* (on a separate line if its length exceeds twenty characters), and the principal's attributes. The maximum ticket lifetime is the maximum period that a Ticket-Granting-Ticket issued to this principal will be valid. Any ticket lifetime up to this value can be requested using an option on the **k4init** command. The key version is an integer set to one when the principal is created and incremented each time the password is changed. The principal's expiration date is displayed in local time, based on the setting of the TZ environment variable.

Exit Values

0 Indicates the successful completion of the command. No output is produced for principal names that do not exist.

1 Indicates that an error occurred and no principal was listed. One of the following conditions was detected:

- The command was incorrectly specified with too many operands, conflicting flags, or a flag that is not valid.
- The host on which the command was issued is not an authentication server.
- The **kdb_util** command was unsuccessful.

Security

The **lskp** command can be run by the root user logged in on a Kerberos server host. It can be invoked indirectly as a Sysctl procedure by a Kerberos database administrator who has a valid ticket and is listed in the **admin_acl.get** file.

Location

/usr/kerberos/etc/lskp

Related Information

Commands: **chkp**, **kadmin**, **kdb_edit**, **mkkp**, **rmkp**, **sysctl**

Examples

1. To list the predefined Kerberos principals, enter:

```
lskp -p
```

You should receive output similar to the following:

```
krbtgt.ABC.DEF.GHI.COM

                    tkt-life: 30d   key-vers: 1

                    expires: 2037-12-31 23:59


default             tkt-life: 30d   key-vers: 1

                    expires: 2037-12-31 23:59


changepw-kerberos   tkt-life: 30d   key-vers: 1

                    expires: 2037-12-31 23:59


K.M                 tkt-life: 30d   key-vers: 1

                    expires: 2037-12-31 23:59
```

2. To list two specific Kerberos principals, joe.admin and lisa, enter:

```
lskp joe.admin lisa
```

You should receive output similar to the following:

```
joe.admin           tkt-life: 15d+08:46   key-vers: 1

                    expires: 2005-03-15 23:59


lisa                tkt-life: 08:00   key-vers: 1

                    expires: 1997-06-09 23:59
```

lsvsd

Purpose

lsvsd – Displays configured virtual shared disks and their characteristics.

Syntax

lsvsd [−**l**] [−**s**] [−**i**] [*vsd_name*...]

Flags

−l Lists the name of the virtual shared disk, the minor number, the state, the current server node number, and, at the server only, the major and minor number of the logical volume. (This is lowercase **l**, as in **l**ist.)

The state field can have one of the following values:

STP Stopped
SUS Suspended
ACT Active

An asterisk (*) in front of any of these values indicates that the virtual shared disk has been fenced from this node.

This flag is not compatible with the −**s** flag.

The *server_list* of the virtual shared disk is listed.

−s Lists usage statistics about the virtual shared disks. It lists the number of local logical read and write operations, the number of remote logical read and write operations, the number of client logical read and write operations, the number of physical reads and writes, the number of cache hits for read, and the number of 512-byte blocks read and written. The number of blocks read and written is cumulative, so issue **ctlvsd** −**V** to reset this count before measuring it.

The local logical operations are requests which were made by a process executing at the local node, whereas the remote logical operations were made by a process executing on a remote node. *Client operations* are those local logical requests that cannot be satisfied locally, and have to be sent to a remote node. *Physical* operations are those server operations which must be passed to the underlying disk device. Cache read hits are those server reads which do not require a device read, because the read operation was satisfied from the virtual shared disk cache.

This flag is not compatible with the −**l** flag.

−i Lists the “node to IP address” map that is currently used by the IBM Virtual Shared Disk driver.

Operands

node_number
Specifies a node.

ip_address Specifies an IP address.

switch_number
Specifies a switch node number.

vsd_name Specifies a virtual shared disk.

Description

The **lsvsd** command displays information about virtual shared disks currently configured on the node on which the command is run. If a list of virtual shared disks follows the flags, information about those virtual shared disks is displayed. **lsvsd** with no arguments or flags lists the names of all the virtual shared disks currently configured on the node.

The **lsvsd** command displays information about both the configuration and the usage of a virtual shared disk.

You can use the System Management Interface Tool (SMIT) to run the **lsvsd** command. To use SMIT, enter:

```
smit vsd_mgmt
```

and select the Show All Managed Virtual Shared Disk Characteristics option.

Prerequisite Information

PSSP: Managing Shared Disks

Location

/usr/lpp/csd/bin/lsvsd

Related Information

Commands: **cfgvsd**, **ctlvsd**, **preparevsd**, **resumevsd**, **startvsd**, **stopvsd**, **suspendvsd**, **ucfgvsd**, **updatevsdnode**, **updatevsdtab**

Examples

1. To list all virtual shared disks in the system, enter:

   ```
   lsvsd
   ```

   The system displays a message similar to the following:

   ```
   vsd00

   vsd01

   :
   ```

2. To list virtual shared disks and their characteristics, enter:

   ```
   lsvsd -l
   ```

   The system displays a message similar to the following:

   ```
   minor  state  server  lv_major  lv_minor  vsd_name  option   size (MB)

   83     STP      -1        0         0     vsdn08v3  cache       20

   84     STP      -1        0         0     vsdn08v4  nocache     16
   ```

3. To list statistics about virtual shared disks and precede the column output with a header, enter:

```
lsvsd -s
```

The system displays a message similar to the following:

```
lc-rd lc-wt rm-rd rm-wt c-rd c-wt  p-rd  p-wt  h-rd   br   bw  vsd_name

   84    84  2858   169    0    0   348   253  2605  164  184  vsd.vsd1

    0     0     0     0    0    0     0     0     0    0    0  vsd.rl01

    0     0     0     0    0    0     0     0     0    0    0  vsd.rl02
```

4.

Issuing **lsvsd** –**i** when VSD/KLAPI is enabled produces output similar to:

```
node        IP address

  1         9.114.43.129

  3         9.114.43.131

  5         9.114.43.133

  7         9.114.43.135

  9         KLAPI[   8]

 10         9.114.43.138

 11         KLAPI[  10]

 12         9.114.43.140

 13         9.114.43.141

 14         9.114.43.142

 15         9.114.43.143

 16         9.114.43.144
```

The following table spells out the names of the headers used in the displays for the **-l** and **-s** options:

Header	Meaning
`minor`	Virtual shared disk minor number
`state`	State of this virtual shared disk:*active, stopped, suspended*
`server`	Primary node for this virtual shared disk
`lv major`	Logical volume major number
`lv minor`	Logical volume minor number
`vsd_name`	Name of this virtual shared disk
`option`	Option:*cache* or *nocache*
`lc-rd`	Local logical reads
`lc-wt`	Local logical writes
`rm-rd`	Remote logical reads
`rm-wt`	Remote logical writes
`c-rd`	Client logical reads
`c-wt`	Client logical writes

Header	Meaning
p-rd	Physical reads
p-wt	Physical writes
h-rd	Reads from cache
br	Blocks read
bw	Blocks written

mkamdent

Purpose

mkamdent – Creates user home directory entries in the **/u** automounter map files.

Syntax

mkamdent [–**s** *server_path*] *user_names*

Flags

–**s** *server_path*
Specifies the location from which the users' home directory is served. The format is *server_name:base_path*. If this flag is not specified, the default values will be taken from the SP site environment variables **homedir_server** for the *server_name* and **homedir_path** for the *base_path*. These environment variables are set using the **spsitenv** command.

Operands

user_names Specifies a list of users to add to the source file, separated by spaces.

Description

Use this command to create user home directory entries in the **/u** automounter map files. Typically, user home directory entries are generated by the SP User Management Services when a new user is added to the system. However, if SP User Management Services are turned off and SP Automounter Support is still turned on, this command can be used to add user entries to the automounter **/u** map. This command can also be used to add automounter support for preexisting users that were not added using SP User Management Services and for **/u** subdirectories that are not associated with SP users.

Files

/etc/auto/maps/auto.u
The default **/u** automounter map file.

/etc/amd/amd-maps/amd.u
The default **/u** map file used by the Amd automounter on PSSP 2.2 and older nodes.

Security

You must have root privilege to run this command.

Location

/usr/lpp/ssp/bin/mkamdent

Related Information

The “Managing the Automounter” and “Managing User Accounts” chapters in *PSSP: Administration Guide*.

Commands: **spsitenv**

Examples

To create automounter entries in the **/u** map file for multiple users, enter:

```
mkamdent -s hostx:/home/hostx john ken pat paul ron
```

This assumes the following directories already exist on **hostx**:

- **/home/hostx/john**
- **/home/hostx/ken**
- **/home/hostx/pat**
- **/home/hostx/paul**
- **/home/hostx/ron**

mkautomap

Purpose

mkautomap – Generates an equivalent Automount map file from an Amd map file.

Syntax

mkautomap [–**n**] [–**o** *Automount_map*] [–**f** *filesystem*] [*Amd_map*]

Flags

–**n** Specifies that an entry for the Automount map should not be added to the **/etc/auto.master** master map file.

–**o** *Automount_map*
Specifies the file name of the Automount map file in which the generated output will be placed. If *Automount_map* does not exist, it will be created. If it does exist, it will be replaced. If this flag is not specified, *Automount_map* will default to **/etc/auto/maps/auto.u**.

–**f** *filesystem*
Specifies the name of the file system associated with the automounter map files. If this flag is not specified, the file system will default to **/u**.

Operands

Amd_map Specifies the file name of the Amd map file that is used as input for generating the Automount map file. If *Amd_map* does not exist, an error will occur. If this option is not specified, *Amd_map* will default to **/etc/amd/amd-maps/amd.u**.

Description

The **mkautomap** command is a migration command used to generate an Automount map file from the Amd map file *Amd_map* created by a previous SP release. Only Amd map file entries created by a previous SP release will be recognized. If the Amd map file was modified by the customer, results may be unpredictable. If an Amd map entry cannot be properly interpreted, a message will be written to standard error, and that entry will be ignored. Processing will continue with the next map entry. All recognized entries will be interpreted and equivalent Automount map entries will be written to a temporary file *Automount_map***.tmp**. If no errors were encountered during processing, the temporary file will be renamed to *Automount_map*.

If all Amd map entries were successfully generated into Automount map entries and written to *Automount_map*, the **/etc/auto.master** Automount master file will be updated unless the –**n** flag is specified. A master map file entry associating the *filesystem* with the *Automount_map* will be added. Also, any default mount options specified in *Amd_map* will be added to the master map file entry for *filesystem*. This master map file entry will be appended to **/etc/auto.master** and if the file does not exist, it will be created.

Files

/etc/amd/amd-maps/amd.u
The default Amd map file used as input to this command.

/etc/auto/maps/auto.u
The default Automount map file generated as output from this command.

/etc/auto/maps/auto.u.tmp
The default temporary Automount map file containing all successfully generated Automount entries. This file will only remain after command execution if errors occurred while processing some Amd map file entries.

/etc/auto.master
The Automount master map file which contains a list of all directories controlled by the automount daemon and their corresponding map files and default mount options.

| ## Security

| You must have root privilege to run this command.

Restrictions

Use this command only with **amd.u** map files created by PSSP User Management Services. Using other Amd map files or modified **amd.u** map files as input to this command, will produce unpredictable results.

Related Information

The "Migrating to the Latest Level of PSSP" chapter in *PSSP: Installation and Migration Guide*

The "Managing the Automounter" chapter in *PSSP: Administration Guide*

Location

/usr/lpp/ssp/install/bin/mkautomap

Examples

To create the SP Automount **/u** map file from the Amd map file generated by a previous SP release, enter:

```
mkautomap
```

mkconfig

Purpose

mkconfig – Creates the **config_info** file for each of the boot/install server's clients on the server.

Syntax

mkconfig

Flags

None.

Operands

None.

Description

Use this command to make the **config_info** files for all the clients of a boot/install server if the client is not set to boot from disk. The **mkconfig** command is intended to run only on the server node. This command creates a **config_info** file named **/tftpboot/***host_name***.config_info** for each client node.

Standard Error

This command writes error messages (as necessary) to standard error.

Exit Values

0 Indicates the successful completion of the command.

–1 Indicates that an error occurred.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Location

/usr/lpp/ssp/bin/mkconfig

Related Information

Commands: **setup_server**

Examples

To make the **config.info** files for all boot/install clients of a server, enter on the server:

```
mkconfig
```

mkinstall

Purpose

mkinstall – Creates the **install_info** file for each of the server's clients on the server.

Syntax

mkinstall

Flags

None.

Operands

None.

Description

Use this command on the server node to make the **install_info** files for all clients of a boot/install server. The **mkinstall** command creates a **/tftpboot/***host_name***.install_info** file for each client node.

Standard Error

This command writes error messages (as necessary) to standard error.

Exit Values

0 Indicates the successful completion of the command.

–1 Indicates that an error occurred.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Location

/usr/lpp/ssp/bin/mkinstall

Related Information

Commands: **setup_server**

Examples

To make the **install.info** files for all boot/install clients of a server, enter on the server:

```
mkinstall
```

mkkp

Purpose

mkkp – Makes Kerberos Version 4 principals.

Syntax

mkkp –**h**

mkkp [–**e** *expiration*] [–**l** *lifetime*] *name*[.*instance*] ...

Flags

–**h** Displays usage information.

–**e** *expiration*
Specifies the expiration date for new principals. If omitted, the expiration date is set to the value assigned to the principal named **default**. The date must be entered in the format *yyyy-mm-dd* and the year must be a value from 1970 to 2037. The time of expiration is set to 11:59 PM local time on the date specified.

–**l** *lifetime* Specifies the maximum ticket lifetime for new principals. If omitted, the maximum ticket lifetime is set to the value assigned to the principal named **default**. The lifetime must be specified as a decimal number from 0 to 255. These values correspond to a range of time intervals from five minutes to 30 days. Refer to *PSSP: Administration Guide* for a complete list of the possible ticket lifetime values you can enter and the corresponding durations in days, hours, and minutes. The following list shows a representative sample with approximate durations:

```
lifetime operand - Approximate duration

     141               1 day

     151               2 days

     170               1 week

     180               2 weeks

     191               1 month
```

Operands

name[.*instance*] **...**
Identifies the principals to add to the Kerberos authentication database.

Description

Use this command to create principals in the Kerberos Version 4 database on the local host. It allows the default values for the expiration date and maximum ticket lifetime to be overridden. Principals created in this way have no passwords. Before a user can **k4init** as the new principal, an administrator must set your initial password using the **kpasswd**, **kadmin**, or **kdb_edit** command directly. This command should normally be used only on the primary server. If there are secondary authentication servers, the **push-kprop** command is invoked to propagate the change to the other servers. The command can be used to update a secondary server's database, but the changes may be negated by a subsequent update from the primary.

Files

/var/kerberos/database/admin_acl.add
Access control list for **kadmin**, **mkkp**, and **rmkp**.

/var/kerberos/database/principal.*
Kerberos database files.

Exit Values

0 Indicates the successful completion of the command. All specified principals that did not already exist were created. If you specified a principal that exists, a message is written to standard error and processing continues with any remaining principals.

1 Indicates that an error occurred and no principal was added. One of the following conditions was detected:

- The command was incorrectly specified with no operand or a flag that is not valid.
- The host on which the command was issued is not an authentication server.
- The **kdb_edit** or **kdb_util** command was unsuccessful.

Security

The **mkkp** command can be run by the root user logged in on a Kerberos server host. It can be invoked indirectly as a Sysctl procedure by a Kerberos database administrator who has a valid ticket and is listed in the **admin_acl.add file**.

Location

/usr/kerberos/etc/mkkp

Related Information

Commands: **chkp**, **kadmin**, **kdb_edit**, **kpasswd**, **lskp**, **rmkp**, **sysctl**

Examples

The following example adds two principals to the database. Both principals are set to expire 30 June 2005. The default value for the maximum ticket lifetime is used.

```
mkkp -e 2005-06-30 kelly kelly.admin
```

mknimclient

Purpose

mknimclient – Makes a node a Network Installation Management (NIM) client of its boot/install server.

Syntax

mknimclient −**h** | −**l** *node_list*

Flags

−**h** Displays usage information. If the command is issued with the −**h** flag, the syntax description is displayed to standard output and no other action is taken (even if other valid flags are entered along with the −**h** flag).

−**l** *node_list* Indicates by *node_list* the SP nodes to be configured as clients of their boot/install servers. The *node_list* is a comma-separated list of node numbers.

Operands

None.

Description

Use this command to define a node as a NIM client. This is accomplished by determining the node's boot/install server from the System Data Repository (SDR) and configuring that client node as a NIM client on that server. When complete, the NIM configuration database on the server contains an entry for the specified client.

Notes:

1. This command results in no processing on the client node.
2. The assignment of a boot/install server for a node must first be made using **spbootins**.

Standard Error

This command writes error messages (as necessary) to standard error.

Exit Values

0 Indicates the successful completion of the command.

−**1** Indicates that an error occurred.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Location

/usr/lpp/ssp/bin/mknimclient

Related Information

Commands: **delnimclient**, **setup_server**

Examples

To define nodes 1, 3, and 5 as NIM clients of their respective boot/install servers, enter:

```
mknimclient -l 1,3,5
```

mknimint

Purpose

mknimint – Creates the necessary Network Installation Management (NIM) interfaces on a NIM master.

Syntax

mknimint −**h** | −**l** *node_list*

Flags

−**h** Displays usage information. If the command is issued with the −**h** flag, the syntax description is displayed to standard output and no other action is taken (even if other valid flags are entered along with the −**h** flag).

−**l** *node_list* Indicates by *node_list* the SP nodes on which to perform this operation. The *node_list* is a comma-separated list of node numbers. These nodes should have been previously configured as NIM masters (see the **mknimmast** command).

Operands

None.

Description

Use this command to define to NIM new Ethernet network adapters and interfaces on the control workstation and boot/install servers. On the control workstation, any networks not previously defined are defined and NIM interfaces added. On a boot/install server, all the Ethernet networks and interfaces are defined; it then defines all token ring and Ethernet networks that are known on the control workstation (with the **netstat** −**ni** command) and defines interfaces for them as well. This is so that resources like the **lppsource** can be served from the control workstation to a client node by the boot/install server if the client and control workstation are on the same subnetwork.

To serve a resource to a client that is not on the same subnetwork as the control workstation, routing is required. Routing is done in **mknimclient**.

Standard Error

This command writes error messages (as necessary) to standard error.

Exit Values

0 Indicates the successful completion of the command.

−**1** Indicates that an error occurred.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Location

/usr/lpp/ssp/bin/mknimint

Related Information

Commands: **setup_server**

Examples

To make NIM interface definitions for nodes 1, 3, and 5, enter:

```
mknimint -l 1,3,5
```

mknimmast

Purpose

mknimmast – Configures a node as a Network Installation Management (NIM) master.

Syntax

mknimmast –**h** –**l** *node_list*

Flags

–**h** Displays usage information. If the command is issued with the –**h** flag, the syntax description is displayed to standard output and no other action is taken (even if other valid flags are entered along with the –**h** flag).

–**l** *node_list* Indicates by *node_list* the SP nodes to be configured as NIM masters. The *node_list* is a comma-separated list of node numbers.

Operands

None.

Description

Use this command to define a boot/install server node as a NIM master for the subsequent installation of client nodes. It verifies that the listed nodes are defined as boot/install servers in the System Data Repository (SDR). It then installs the NIM master AIX file sets and configures the nodes as NIM masters.

Standard Error

This command writes error messages (as necessary) to standard error.

Exit Values

0 Indicates the successful completion of the command.

–**1** Indicates that an error occurred.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Location

/usr/lpp/ssp/bin/mknimmast

Related Information

Commands: **delnimmast**, **setup_server**

Examples

To define nodes 1, 3, and 5 as NIM masters, enter:

```
mknimmast -l 1,3,5
```

mknimres

Purpose

mknimres – Creates the necessary Network Installation Management (NIM) resources on a NIM master.

Syntax

mknimres −**h** | −**l** *node_list*

Flags

−**h** Displays usage information. If the command is issued with the −**h** flag, the syntax description is displayed to standard output and no other action is taken (even if other valid flags are entered along with the −**h** flag).

−**l** *node_list* Indicates by *node_list* the SP nodes on which to perform this operation. The *node_list* is a comma-separated list of node numbers. These nodes should have been previously configured as NIM masters (see **mknimmast**).

Operands

None.

Description

Use this command to make all the NIM resources for installation, diagnostics, migration, and customization. No resources are allocated to client nodes. The set of resources needed is determined from the list of client nodes found in the System Data Repository (SDR) for the *node_list*. Any required AIX install and **mksysb** images are defined as NIM resources. For boot/install server nodes, NIM Shared Product Object Tree (SPOT) directories are created and **mksysb** images are copied, as required. Because of the large data volumes required for SPOTs and install images, all checking is done before copying data.

Creation of the NIM lppsource resource on a boot/install server will result in **setup_server** creating a lock in the lppsource directory on the control workstation.

Standard Error

This command writes error messages (as necessary) to standard error.

Exit Values

0 Indicates the successful completion of the command.

−**1** Indicates that an error occurred.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Location

/usr/lpp/ssp/bin/mknimres

Related Information

Commands: **setup_server**

Examples

To make NIM resources for boot/install servers 1, 3, and 5, enter:

```
mknimres -l 1,3,5
```

monitorvsd

Purpose

monitorvsd – Enables, disables, or lists the virtual shared disks that will be monitored.

Syntax

monitorvsd −**d** *vsd_name ...* | −**e** *vsd_name ...* |
−**E** | −**D** | −**l**

Flags

−**d** *vsd_name ...*	Disables monitoring the specified shared disks. The *vsd_names* are space separated.
−**e** *vsd_name ...*	Enables monitoring the specified shared disks.
−**E**	Enables monitoring all shared disks that were previously enabled with the **-e** flag.
−**D**	Disables all monitoring.
−**l**	Lists the virtual shared disks that are being monitored.

Operands

None.

Description

The **monitorvsd** command enables and disables virtual shared disks to be monitored by the PSSP Event Management services. In particular, the statistics that are returned by the **lsvsd -s** command are made available to Event Management.

Monitoring can be enabled for a maximum of 300 virtual shared disks on a given node.

Security

You must have root privilege to run this command.

Prerequisite Information

PSSP: Managing Shared Disks

For information on the Event Management services, refer to "The Event Management Subsystem" and "Using the Problem Management Subsystem" chapters in *PSSP: Administration Guide*.

Location

/usr/lpp/csd/bin/monitorvsd

Related Information

Commands: **cfgvsd**, **lsvsd**, **spevent**

Examples

1. To enable monitoring the "vsd1n1" shared disk, enter:

   ```
   monitorvsd -e vsd1n1
   ```

   The system displays a message similar to the following:

   ```
   monitorvsd: Enabled: vsd1n1
   ```

2. To list all monitored virtual shared disks on a node, enter:

   ```
   monitorvsd -l
   ```

   The system displays a message similar to the following:

   ```
    vsd1n1

       vsd2n1

       vsd9n1
   ```

3. To disable all monitoring, enter:

   ```
   monitorvsd -D
   ```

mult_senders_test

Purpose

mult_senders_test – Detects nodes that are injecting damaged packets into the switch network.

Attention

ATTENTION – READ THIS FIRST: Do **not** activate the SP Switch advanced diagnostic facility until you have read this section completely, and understand this material. If you are not certain how to properly use this facility, or if you are not under the guidance of IBM Service, do **not** activate this facility.

Activating this facility may result in degraded performance of your system. Activating this facility may also result in longer response times, higher processor loads, and the consumption of system disk resources. Activating this facility may also obscure or modify the symptoms of timing-related problems.

Syntax

mult_senders_test –**r** *receiver* [–**g**] [–**m** *model*] [–**t** *max_time*]
[–**a** *allowed_sender(s)*] [–**f** *forbidden sender(s)*]
[–**A** *allowed_senders_file*] [–**F** *forbidden_senders_file*]
[–**z** *data_size*] [–**p** *pattern_file(s)*] [–**h**]

Flags

–**r** *receiver* Specifies a receiver node ID (or name).

–**m** *model* Specifies a test model that will be used for testing. *model* is the name of the model to be used.

–**t** *max_time*
Specifies maximal execution time.

–**a** *allowed_sender(s)*
Specifies a list of nodes that the test can use. *allowed_senders* is a blank–separated list of node identifiers. A node identifier can be a host name, IP address, frame,slot pair, or node number.

–**f** *forbidden_sender(s)*
Specifies a list of nodes that the test cannot use. *forbidden_sender* is a blank–separated list of node identifiers.

–**A** *allowed_senders_file*
Specifies a file containing the list of nodes that the test can use. *allowed_senders_file* is a path to a file that contains a list of node identifiers.

–**F** *forbidden_senders_file*
Specifies a file containing the list of nodes that the test cannot use. *forbidden_senders_file* is a path to a file that contains a list of node identifiers.

–**z** *data_size*
: Specifies an amount of data, in MB, to be sent in every single test iteration by each sender.

–**p** *pattern_file(s)*
: Specifies a list of paths to the pattern files. *pattern_files* is a blank–separated list of paths. Each pattern file path is a full path to a file accessible from each participating node.

–**g**
: Request to use SPD GUI.

–**h**
: Request usage information be displayed.

Operands

None.

Description

This command starts the multiple senders test, which will find the malfunctioning sender(s) among a specified group of nodes or among the whole partition. You are required to specify the receiver that reported the "bad packet" error by node ID, hostname or IP address.

Primary and Backup nodes cannot participate in the test as receiver(s) or sender(s). If you specify Primary or Backup nodes as receiver(s), test will exit and an error message will be displayed.

The model argument lets you select a test model. By default the "All available nodes are senders" model is selected (this is the only supported model).

You can specify the nodes that are allowed to participate in the test, or nodes that are not allowed to participate in the test. If the same node is present in both lists, it is not allowed to participate in the test. You must be aware that the selected nodes will not be able to run any application that uses a switch network during the test execution. By default all nodes are allowed to participate in the test. These nodes could be specified as a list of nodes or as a file that contains the list. The *data_size* argument allows you to control the amount of data that will be sent by every sender on every test iteration. By default this value is set to 360MB.

You can provide a path to a file that contains the data pattern to be used during the test. By default the output of the test is displayed on the command line. You can request to display the output on the SPD GUI.

Location

/usr/lpp/ssp/bin/spd/mult_senders_test

Examples

1. To execute multiple senders test using receiver node #11 enter:

   ```
   mult_senders_test –r 11
   ```

2. To execute multiple senders test using receiver node n01 and specifying allowed nodes by host name, enter:

   ```
   mult_senders_test –r n01–a n05 n06 n11
   ```

3. To execute multiple senders test using receiver node n01, specifying a forbidden node by frame,slot, enter:

```
mult_senders_test -r n01 -f 2,9
```

4. To execute model A of multiple senders test, enter:

```
mult_senders_test -r 11 -m ModelA
```

5. To increase amount of data sent through by each sender to the receiver enter:

```
mult_senders_test -r 11 -z 1000
```

6. To use a different data pattern, create a data file, make it accessible to nodes (copy to every node or mount using the same name), enter:

```
mult_senders_test -r 11 -p /tmp/spd/pattern1.dat
```

ngaddto

Purpose

ngaddto – Adds nodes and node groups to the definition list of the destination node group.

Syntax

ngaddto [−**h**] | [−**G**] *dest_nodegroup*
nodenum | *nodegroup* [*nodenum* | *nodegroup*] ...

Flags

−**h** Displays usage information.

−**G** Specifies that the destination node group is global.

Operands

dest_nodegroup
Specifies the node group to receive the new additions.

nodenum Specifies a node to add to the definition list of the destination node group. This is supplied as a space-delimited list of node numbers.

nodegroup Specifies a named node group to add to the definition list of the destination node group. Node groups are given as a space-delimited list. Node numbers and node group names being added to the destination node group can be intermixed.

Description

Use this command to add nodes and node groups to the definition list of the destination node group. If the −**G** flag is specified, the destination node group must be global. If the −**G** flag is not specified, the destination node group must belong to the current system partition. If the destination node group does not exist, you will receive an error. If the destination node group or *nodegroup* is a name that is not valid, you will receive an error. Nodes and node groups that do not currently exist can be added to the destination node group. When the node group is resolved by the **ngresolve** command, nonexistent members are ignored.

Exit Values

0 Indicates the successful completion of the command.

nonzero Indicates that an error occurred.

Security

You must have write access to the SDR to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

Refer to the “Managing Node Groups” chapter in *PSSP: Administration Guide* for additional node grouping information.

Location

/usr/lpp/ssp/bin/ngaddto

Related Information

Commands: **ngcreate**, **ngdelete**, **ngdelfrom**, **ngfind**, **nglist**, **ngnew**, **ngresolve**

Examples

1. To add nodes 1 and 3 and node group ngb to the definition list of node group nga, enter:

   ```
   ngaddto nga 1 3 ngb
   ```

2. To add nodes 1 and 16 and global node group g2 to the global definition list of node group g1, enter:

   ```
   ngaddto -G g1 1 16 g2
   ```

ngclean

Purpose

ngclean – Cleans up a node group, removing references to nodes and node groups that are not in the current system partition. Node groups with empty definition lists will be deleted.

Syntax

ngclean [−**h**] | [−**G**] [−**r**] {−**a** | *nodegroup* [*nodegroup* ...]}

Flags

−**h** Displays usage information.

−**a** Cleans up all node groups in the current system partition or all system-wide node groups if the −**G** flag is also specified.

−**r** Does not modify node groups. Issues a report on how node groups would be affected by running this command (without the −**r** option).

−**G** Examines global node groups.

Operands

nodegroup Specifies the node groups to be cleaned. If the −**a** flag is provided, all node groups will be cleaned and no node groups should be specified.

Description

Use this command to examine node group definition lists and to remove references to nodes and node groups that do not exist in the current system partition or the SP system if −**G** is supplied. Node groups with empty definition lists will be deleted. If the −**r** flag is specified, the nodes and node groups will not be removed, but a report will be generated.

Exit Values

0 Indicates the successful completion of the command.

nonzero Indicates that an error occurred.

| ## Security

| You must have write access to the SDR to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

Refer to the “Managing Node Groups” chapter in *PSSP: Administration Guide* for additional node grouping information.

Location

/usr/lpp/ssp/bin/ngclean

Related Information

Command: **ngaddto**, **ngcreate**, **ngdelete**, **ngdelfrom**, **ngfind**, **nglist**, **ngnew**, **ngresolve**

Examples

1. To clean up all system node groups, enter:

```
ngclean -Ga
```

2. To clean up the node group my.ng in the current system partition, enter:

```
ngclean my.ng
```

ngcreate

Purpose

ngcreate – Creates and optionally populates a named node group.

Syntax

ngcreate [–**h**] | [–**s** *frame_range*:*slot_range*] [–**n** *node_range*]
[–**w** *host_name,host_name*, ...] [–**e** *host_name,host_name*, ...]
[–**N** *nodegroup,nodegroup*, ...] [–**a**] [–**G**] *dest_nodegroup*

Flags

–h Displays usage information.

–s Specifies a range of frames and slots on each frame to add to the node group.

–n Specifies a range of nodes to be added to the node group.

–w Specifies a comma-delimited list of hosts to add to the node group.

–a Specifies that all nodes in the current system partition be added to the node group. If the –**G** flag is also provided, all nodes in the SP system are included.

–e Specifies a comma-delimited exclusion list. These hosts are not added to the node group even if they are specified by another option.

–N Specifies a comma-delimited list of node groups to add to this node group.

–G Creates a global node group. System partition boundaries are ignored.

Operands

dest_nodegroup
Specifies the name associated with the node group being created.

Description

Use this command to create a node group named *dest_nodegroup*. The destination node group is populated based on the supplied options. Node group names must begin with a letter and can be followed by any letters or numbers, a period (.), or an underscore (_). If the destination node group already exists, you will receive an error.

Exit Values

0 Indicates the successful completion of the command.

nonzero Indicates that an error occurred.

Security

You must have write access to the SDR to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

Refer to the "Managing Node Groups" chapter in *PSSP: Administration Guide* for additional node grouping information.

Location

/usr/lpp/ssp/bin/ngcreate

Related Information

Commands: **ngaddto**, **ngdelete**, **ngdelfrom**, **ngfind**, **nglist**, **ngnew**, **ngresolve**

Examples

To create a node group called sample_ng that contains all the nodes in the current system partition except for k22n01, enter:

```
ngcreate -ae k22n01 sample_ng
```

ngdelete

Purpose

ngdelete – Removes node groups from persistent storage.

Syntax

ngdelete [−**h**] | [−**u**] [−**G**] *nodegroup* [*nodegroup* ...]

Flags

−h Displays usage information.

−u Removes the *nodegroup*, but leaves references to this *nodegroup* in the definition list of any any node group that contains it.

−G Specifies that the *nodegroup* is global.

Operands

nodegroup Specifies the name of the node group to be deleted.

Description

Use this command to remove node groups from persistent storage. By default, the node group is removed from any node group that contains it. If the −**u** flag is specified, references to this deleted node group will remain in containing node groups.

Exit Values

0 Indicates the successful completion of the command.

nonzero Indicates that an error occurred.

| ## Security

| You must have write access to the SDR to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

Refer to the "Managing Node Groups" chapter in *PSSP: Administration Guide* for additional node grouping information.

Location

/usr/lpp/ssp/bin/ngdelete

Related Information

Commands: **ngaddto**, **ngcreate**, **ngdelfrom**, **ngfind**, **nglist**, **ngnew**, **ngresolve**

Examples

To delete nodegroups ngc and ngd, enter:

```
ngdelete ngc ngd
```

ngdelfrom

Purpose

ngdelfrom – Deletes nodes and node groups from the definition list of the destination node group.

Syntax

ngdelfrom [−**h**] | [−**G**] *dest_nodegroup* *nodenum* | *nodegroup* [*nodenum* | *nodegroup*] ...

Flags

−h Displays usage information.

−G Specifies that the *dest_nodegroup* is global.

Operands

dest_nodegroup
Specifies the node group to be modified.

nodenum Specifies a node to remove. Nodes are specified as a space-delimited list of node numbers.

nodegroup Specifies a named node group to remove. Node groups are specified as a space-delimited list of node group names. Only the node group name will be removed from the destination nodegroup. The group will not be resolved into an individual list of nodes.

Note: Nodes numbers and node group names being removed can be intermixed.

Description

Use this command to remove nodes and node groups from the definition list of the destination node group. If the −**G** flag is specified, the *dest_nodegroup* must be global. If the −**G** flag is not specified, the *dest_nodegroup* must belong to the current system partition.

Exit Values

0 Indicates the successful completion of the command.

nonzero Indicates that an error occurred.

Security

| You must have write access to the SDR to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

Refer to the “Managing Node Groups” chapter in *PSSP: Administration Guide* for additional node grouping information.

Location

/usr/lpp/ssp/bin/ngdelfrom

Related Information

Commands: **ngaddto**, **ngcreate**, **ngdelete**, **ngfind**, **nglist**, **ngnew**, **ngresolve**

Examples

To remove node 5 and node group ngc from nga, enter:

```
ngdelfrom nga 5 ngc
```

ngfind

Purpose

ngfind – Returns a list of all node groups whose definition list contains the specified node or node group.

Syntax

ngfind [−**h**] | [−**G**] *nodegroup* | *node*

Flags

−**h**	Displays usage information.
−**G**	Returns all global node groups that contain the specified global node group or node in their definition list. The default scope is the current system partition.

Operands

nodegroup	Searches node group definition lists for references to this node group.
node	Searches node group definition lists for references to this node.

Description

Use this command to list all node groups that contain the specified node or node group in their definition list. If the specified node or node group does not exist in a node group definition list, no node groups will be listed and the command will complete successfully. Use this command to determine what other node groups would be affected by changes to the specified node group.

Exit Values

0	Indicates the successful completion of the command.
nonzero	Indicates that an error occurred.

Security

You must have write access to the SDR to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

Refer to the "Managing Node Groups" chapter in *PSSP: Administration Guide* for additional node grouping information.

Location

/usr/lpp/ssp/bin/ngfind

Related Information

Commands: **ngaddto**, **ngcreate**, **ngdelete**, **ngdelfrom**, **nglist**, **ngnew**, **ngresolve**

Examples

To display a list of all node groups that contain node group test_B, enter:

```
> ngfind test_B

test_A

test_D
```

nglist

Purpose

nglist – Returns a list of all node groups in the current system partition.

Syntax

nglist [–**h**] | [–**G**]

Flags

–h Displays usage information.

–G Returns all global node groups.

Operands

None.

Description

Use this command to list all node groups in the current system partition to standard output. If the –**G** flag is specified, it will list all system node groups.

Standard Output

A list of node groups is written to standard output, one node group per line.

Exit Values

0 Indicates the successful completion of the command.

nonzero Indicates that an error occurred.

| ## Security

| You must have write access to the SDR to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

Refer to the "Managing Node Groups" chapter in *PSSP: Administration Guide* for additional node grouping information.

Location

/usr/lpp/ssp/bin/nglist

Related Information

Commands: **ngaddto**, **ngcreate**, **ngdelete**, **ngdelfrom**, **ngfind**, **ngnew**, **ngresolve**

Examples

1. To display a list of all node groups in the current system partition, enter:

```
> nglist

nga

ngb

sampleng

test_A
```

2. To display a list of all global node groups, enter:

```
> nglist -G

g1

g2

g3

test_A
```

Note: The global node group test_A is not the same as node group test_A in the current system partition. The global scope and system partition dependent scope are independent name spaces and are stored in separate classes in the System Data Repository (SDR).

ngnew

Purpose

ngnew – Creates but does not populate new node groups in persistent storage.

Syntax

ngnew [–**h**] | [–**G**] *nodegroup* [*nodegroup* ...]

Flags

–h Displays usage information.

–G Creates global node groups.

Operands

nodegroup Specifies the node group to be created.

Description

Use this command to create new node groups. If the *nodegroup* already exists, you will receive an error. A valid node group name must begin with a letter. If the *nodegroup* is not a valid name, you will receive an error. If a node group in the list cannot be successfully created, it will not affect the creation of the other supplied node groups. A nonzero return code is returned.

Exit Values

0 Indicates the successful completion of the command.

nonzero Indicates that an error occurred.

| ## Security

| You must have write access to the SDR to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

Refer to the “Managing Node Groups” chapter in *PSSP: Administration Guide* for additional node grouping information.

Location

/usr/lpp/ssp/bin/ngnew

Related Information

Commands: **ngaddto**, **ngcreate**, **ngdelete**, **ngdelfrom**, **ngfind**, **nglist**, **ngresolve**

Examples

To create node groups called nga, ngb, and ngc, enter:

```
ngnew nga ngb ngc
```

ngresolve

Purpose

ngresolve – Returns a list of hosts in the specified node group.

Syntax

ngresolve [–**h**] | [–**u** | –**n** | –**w** | –**d**] [–**G**] *nodegroup* [*nodegroup* ...]

Flags

–**h** Displays usage information.

–**u** Writes the definition list of *nodegroup*. Node groups contained by *nodegroup* are left unresolved.

–**n** Specifies that nodes are written as node numbers. This is the default.

–**w** Specifies that nodes are written as fully qualified host names.

–**d** Specifies that nodes are written as fully qualified IP addresses.

–**G** Specifies that node groups are global.

Operands

nodegroup Specifies the node group to be resolved.

Description

Use this command to resolve the supplied named node groups into their constituent nodes. Nodes and node groups that are in the supplied node group but do not currently exist, will resolve to an empty list. If the –**u** flag is specified, these nonexistent nodes and node groups will be displayed.

Standard Output

A resolved list of nodes is written to standard output, one node per line.

Exit Values

0 Indicates the successful completion of the command.

nonzero Indicates that an error occurred.

Security

You must have write access to the SDR to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

Refer to the “Managing Node Groups” chapter in *PSSP: Administration Guide* for additional node grouping information.

Location

/usr/lpp/ssp/bin/ngresolve

Related Information

Commands: **ngaddto**, **ngcreate**, **ngdelete**, **ngdelfrom**, **ngfind nglist**, **ngnew**

Examples

1. To display the definition list for node group nga, enter:

   ```
   > ngresolve -u nga

   1

   3

   ngb
   ```

2. To resolve node group nga into its constituent nodes, enter:

   ```
   > ngresolve nga

   1

   3

   6

   8
   ```

3. To resolve node group nga into fully qualified host names, enter:

   ```
   > ngresolve -w nga

   k22n01.ppd.pok.ibm.com

   k22n03.ppd.pok.ibm.com

   k22n06.ppd.pok.ibm.com

   k22n08.ppd.pok.ibm.com
   ```

4. To display the IP addresses of the nodes in node group nga, enter:

   ```
   > ngresolve -d nga

   129.40.157.65

   129.40.157.67

   129.40.157.70

   129.40.157.72
   ```

node_number

Purpose

node_number – Obtains the node number attribute for a node from the ODM.

Syntax

node_number[−**h**] [−**new**]

Flags

−**h**	Display **node_number** command syntax.
−**new**	Sets return code to −1 if **node_number** is equal to a null string.

Operands

None.

Description

This command is used by the PSSP software to determine the node number of an SP node. The PSSP installation process places the node number in the ODM on the node. This command will retrieve that data.

Standard Output

The node number obtained is printed to standard output.

Standard Error

Any errors from the ODM query will be printed to standard error.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates that an error occurred.

Implementation Specifics

This command is part of the IBM Parallel System Support Program (PSSP) Licensed Program Product (LPP).

Location

/usr/lpp/ssp/install/bin/node_number

Examples

To obtain the node number of an SP node, issue the following on that node:

```
node_number
```

```
5
```

nodecond

Purpose

nodecond – Conditions an SP processing node.

Syntax

nodecond [–**G**] [–**n**] [–**s**] *frame_ID slot_ID*

Flags

–G Specifies Global mode. With this flag, the node to be conditioned can be outside of the current system partition.

–n Obtains the Ethernet hardware address instead of doing a network boot.

–s Performs a slow boot (disables fast IPL mode). Specifies that **s1term** will not be opened after an IPL in diagnostic mode.

Operands

frame_ID Specifies the number of the frame containing the node to be conditioned.

slot_ID Specifies the number of the slot containing the node to be conditioned.

Description

Node conditioning is the administrative procedure used to obtain the Ethernet hardware address of an SP processing node or to initiate a network boot of an SP processing node. The Ethernet hardware address is required by SP System Management for the proper configuration of the system. A network boot of the node is required by the System Management installation procedures.

By default, the **nodecond** command initiates a network boot of the node specified by the *frame_ID* and *slot_ID* operands. The specified node must be in the current system partition unless the –**G** flag is also specified. The frame ID is any configured frame number and the slot ID is taken from the set 1 through 16. The command completes when the node has booted to the point of configuring its console. Using –**n**, the **nodecond** command obtains the Ethernet hardware address of the processing node, specified by the *frame_ID* and *slot_ID* operands. The hardware address is written to standard output and the node is left powered off with the keylock in the Normal position. Using –**s**, the **nodecond** command runs with fast IPL disabled, allowing more diagnostic information to be collected. After this slow boot, **s1term** will not open as it does by default.

As the command executes, it writes status information indicating its progress to **/var/adm/SPlogs/spmon/nc/nc.***frame_ID.slot_ID*.

This command uses the SP Hardware Monitor. Therefore, the user must be authorized to access the Hardware Monitor subsystem and, for the frame specified to the command, the user must be granted Virtual Front Operator Panel (VFOP) and S1 (serial port on the node that you can access via the **s1term** command) permission. Since the Hardware Monitor subsystem uses SP authentication services, the user must execute the **k4init** command prior to executing this command. Alternatively, site-specific procedures can be used to obtain the tokens that are otherwise obtained by **k4init**.

Files

/var/adm/SPlogs/spmon/nc
Directory containing **nodecond** status files.

Security

You must have Hardware Monitor "VFOP" access and serial access to run this command.

Location

/usr/lpp/ssp/bin/nodecond

Related Information

Commands: **hmcmds**, **hmmon**, **s1term**

Examples

1. To fetch the Ethernet hardware address of the node in frame 5 in slot 1 and save it in a file, enter:

```
nodecond -n 5 1 > eth_adrr.5.1
```

2. To network boot the node in frame 7 in slot 16, enter:

```
nodecond 7 16
```

nrunacct

Purpose

nrunacct – Runs on each node every night to merge raw accounting data from the login, fee, disk, print, and process subsystems.

Syntax

nrunacct *yyyymmdd*
[**SETUP** | **WTMPFIX** | **CONNECT1** | **CONNECT2** |
PROCESS |
MERGE | **FEES** | **DISK** | **QUEUEACCT** | **CMS** |
USEREXIT | **CLEANUP**]

Flags

SETUP
Moves the active accounting files to working files and restarts the active files.

WTMPFIX
Verifies the integrity of the **wtmp** file and corrects dates if necessary.

CONNECT1
Calls the **acctcon1** command to produce connect session records.

CONNECT2
Converts connect session records into total accounting records (**tacct.h** format).

PROCESS
Converts process accounting records into total accounting records (**tacct.h** format). Filters out the records that belong to processes that were part of a job that had exclusive use of the node and appends a total accounting fee record to the fee file for each of these jobs. Records are identified as belonging to processes that were part of a job that had exclusive use of the node, only if exclusive use accounting was enabled at the time the job ran.

MERGE
Merges the connect and process total accounting records.

FEES
Converts accounting fee file records into total accounting records (**tacct.h** format) and merges them with the connect and process total accounting records.

DISK
Merges disk accounting records with connect, process, and fee total accounting records.

QUEUEACCT
Sorts the queue (printer) accounting records, converts them into total accounting records (**tacct.h** format), and merges them with other total accounting records.

CMS
Produces command summaries and updates the file that records the date each user last logged into the node.

USEREXIT
If the **/var/adm/nsiteacct** shell file exists, calls it at this point to perform site-dependent processing.

CLEANUP
Deletes temporary files and exits.

Operands

yyyymmdd Specifies the date when accounting is to be rerun.

Description

The **nrunacct** command is the main daily accounting shell procedure, for each individual node. Normally initiated by the **cron** daemon, the **nrunacct** command merges the day's raw connect, fee, disk, queuing system (printer), and process accounting data files for the node.

This command has two parameters that must be entered from the keyboard should you need to restart the **nrunacct** procedure. The date parameter, *YYYYMMDD* enables you to specify the date for which you want to rerun the node accounting. The state parameter enables a user with administrative authority to restart the **nrunacct** procedure at any of its states. For more information on restarting **nrunacct** procedures and on recovering from errors, see "Restart Procedure."

The **nrunacct** command protects active accounting files and summary files in the event of runtime errors, and records its progress by writing descriptive messages into the **/var/adm/acct/nite/active***YYYYMMDD* file. When the **nrunacct** procedure encounters an error, it sends mail to users root and adm, and writes standard errors to **/var/adm/acct/nite/accterr**.

The **nrunacct** procedure also creates two temporary files, **lock** and **lock1**, in the directory **/var/adm/acct/nite**, which it uses to prevent two simultaneous calls to the **nrunacct** procedure. It uses the lastdate file (in the same directory) to prevent more than one invocation per day.

The **nrunacct** command breaks its processing into separate, restartable states. As it completes each state, it writes the name of the next state in the **/var/adm/acct/nite/state***YYYYMMDD* file.

Restart Procedure

To restart the **nrunacct** command after an error, first check the **/var/adm/acct/nite/active***YYYYMMDD* file for diagnostic messages, then fix any damaged data files, such as **pacct** or **wtmp**. Remove the lock files and **lastdate** file (all in the **/var/adm/acct/nite** directory, before restarting the **nrunacct** command. You must specify the *YYYYMMDD* parameter if you are restarting the **nrunacct** command. It specifies the date for which the **nrunacct** command is to rerun accounting. The **nrunacct** procedure determines the entry point for processing by reading the **/var/adm/acct/nite/statefile***YYYYMMDD* file. To override this default action, specify the desired state on the **nrunacct** command line.

It is not usually a good idea to restart the **nrunacct** command in the SETUP state. Instead, perform the setup actions manually and restart accounting with the WTMPFIX state, as follows:

```
/usr/lpp/ssp/bin/nrunacct YYYYMMDD WTMPFIX
```

If the nrunacct command encounters and error in the PROCESS state, remove the last **ptacct** file, because it is incomplete.

Files

/var/adm/wtmp Log in/log off history file.

/var/adm/pacct* Process accounting file.

/var/adm/acct/nite/dacct
Disk usage accounting file.

/var/adm/qacct Active queue accounting file.

/var/adm/fee Record of fees charged to users.

/var/adm/acct/nite/ptacct*.mmdd
Summary version of pacct files.

/var/adm/acct/nite/activeYYYYMMDD
The nrunacct message file.

/var/adm/acct/nite/lock*
Prevents simultaneous invocation of **nrunacct**.

/var/adm/acct/nite/lastdate
Contains last date nrunacct was run.

/var/adm/acct/nite/statefileYYYYMMDD
Contains current state to process.

Security

You must have root privilege to run this command.

Restrictions

Access Control: This command should grant execute (x) access only to members of the adm group.

Location

/usr/lpp/ssp/bin/nrunacct

Related Information

Commands: **acctcms**, **acctcom**, **acctcon1**, **acctcon2**, **acctmerg**, **accton**, **acctprc1**, **acctprc2**, **crontab**, **fwtmp**, **nrunacct**,

Daemon: **cron**

Subroutine: **acct**

File format: **acct**, **failedlogin**, **tacct**, **wtmp**

The System Accounting information found in *AIX Version 4 System Management Guide*

Examples

1. To restart a node's system accounting procedures for a specific date, enter a command similar to the following:

   ```
   nohup /usr/lpp/ssp/bin/nrunacct 19950601 2>> \

   /var/adm/acct/nite/accterr &
   ```

   This example restarts **nrunacct** for the day of June 1 (0601), 1995. The **nrunacct** command reads the file **/var/adm/acct/nite/statefile19950601** to find out the state with which to begin. The **nrunacct** command runs in the background (&), ignoring all INTERRUPT and QUIT signals (**nohup**). Standard error output (2) is added to the end (>>) of the **/var/adm/acct/nite/accterr** file.

2. To restart a node's system accounting procedures for a particular date at a specific state, enter a command similar to the following

   ```
   nohup /usr/lpp/ssp/bin/nrunacct 19950601 FEES 2>> \

   /var/adm/acct/nite/accterr &
   ```

   This example restarts the **nrunacct** command for the day of June 1 (0601), 1995, starting with the FEES state. The **nrunacct** command runs in the background (&), ignoring all INTERRUPT and QUIT signals (the **nohup** command). Standard error output (2) is added to the end (>>) of the **/var/adm/acct/nite/accterr** file.

p_cat

Purpose

p_cat – Issues a parallel cat of files.

Syntax

p_cat [–**w** – | *noderange* | '*hostlist args*'] *file_name* *file_name* ...

Flags

The **p_cat** command requires the first flag or parameter on the command line to be a specification of the hosts on which the command is to be executed.

–**w** – Specifies that host names should be read from standard input. Host names can be in any format accepted by **rsh**.

Operands

noderange Indicates a specification via "node number." The node number corresponds to the position of a node in a frame and its slots. A node number indicates frame and slot. For example, frame 1 slot 1 would be referred to by 1. Frame 2 slot 1 would be node number 17, while frame 3 slot 2 would be 34. If a node occupies more than one slot, either node number refers to that node. Node numbers can be specified as ranges such as 1–3, which would refer to frame 1 slots 1–3, or 23–29,50,2, which would refer to frame 2 slots 7–13, frame 4 slot 2, and frame 1 slot 2. This option is valid only for an SP system.

'*hostlist args*'
Specifies flags and arguments to be passed to the **hostlist** command. Hostlist allows several ways of listing hosts based on various criteria. Refer to the **hostlist** command.

Note: To use the working collective file specified by the WCOLL environment variable, you must specify a null string as the first argument. Refer to the **dsh** command for more information about a working collective.

file_name Specifies a list of file names on the hosts to be concatenated to standard output.

Description

The **p_cat** command issues the AIX **cat** command on multiple hosts. **p_cat** uses **dsh** to execute the **cat** command on multiple hosts. The output of the **cat** command is written to standard output.

Files

working collective file
See the **dsh** command.

Security

You must have access to the AIX Secure Remote Commands to run this command.

This command will automatically forward the DCE credentials if K5 is an enabled AIX authentication method and the user of the command has DCE credentials that can be forwarded. The special DCE credentials for root, called the machine or self host principal credentials, cannot be forwarded. To obtain DCE credentials that can be forwarded as a root user, a root user must issue **dce_login -f**.

Location

/usr/lpp/ssp/bin/p_cat

Related Information

SP commands: **dsh**, **pexec**

AIX command: **cat**

Examples

To copy **/.rhosts** from each host1, host2, and host3 to the local **/.rhosts** file (described previously), enter:

```
p_cat -w host1,host2,host3 /.rhosts >> /.rhosts
```

pcp

Purpose

pcp – Specifies a parallel copy of local files and directories to other hosts.

Syntax

pcp [−**w** − | *noderange* | **'** *hostlist args***'**] [−**D**] [−**p**] [−**r**] *localfile_or_dir remotefile_or_dir*

Flags

The **pcp** command requires the first flag or parameter on the command line to be a specification of the hosts on which the command is to be executed.

−**w** − Specifies that host names should be read from standard input. Host names can be in any format accepted by **rsh**.

−**p** Preserves the modification times and modes of the source files in the copies sent to the destination only if the user has root authority or is the owner of the destination. Without this flag, the **umask** command at the destination modifies the mode of the destination file, and the modification time of the destination file is set to the time the file is received.

−**r** Recursively copies subtrees for directories.

−**D** Recursively forwards the DCE credentials for authentication by using the −**F** flag on the **rsh** command. If this flag is not set, the **dsh** command will not forward DCE credentials.

Operands

noderange Indicates a specification via "node number." The node number corresponds to the position of a node in a frame and its slots. A node number indicates frame and slot. For example, frame 1 slot 1 would be referred to by 1. Frame 2 slot 1 would be node number 17, while frame 3 slot 2 would be 34. If a node occupies more than one slot, either node number refers to that node. Node numbers can be specified as ranges such as 1–3, which would refer to frame 1 slots 1–3, or 23–29,50,2, which would refer to frame 2 slots 7–13, frame 4 slot 2, and frame 1 slot 2. This option is only valid for an SP system.

'*hostlist args***'**
Specifies flags and arguments to be passed to the **hostlist** command. Hostlist allows several ways of listing hosts based on various criteria. Refer to the **hostlist** command.

Note: To use the working collective file specified by the WCOLL environment variable, you must specify a null string as the first argument. Refer to the **dsh** command for more information about a working collective.

local_file_or_dir
Contains the name of the file or directory to be copied to the remote hosts. If a full path name is not specified, the file is looked for relative to the user's current working directory.

remote_file_or_dir
Contains the name of the file or directory to which the file or directory is copied.

Description

The **pcp** command copies files from the local host to one or more others in parallel. **pcp** is similar to **rcp** and in fact uses **rcp** via **dsh**. The −**D**, −**r** and −**p** flags are passed to **rcp**.

Note: Since the **pcp** command uses the secure version of **rcp**, your **.klogin** or **.rhosts** files need to be set up to authorize you on each of the nodes to which you are copying a file. Refer to the chapter on security in *PSSP: Administration Guide*. Otherwise, you see:

```
Permission denied
```

messages from the nodes for which you are not authorized.

Security

You must have access to the AIX Secure Remote Commands to run this command.

If you specify the −**D** flag, this command will automatically forward the DCE credentials if K5 is an enabled AIX authentication method and the user of the command has DCE credentials that can be forwarded. The special DCE credentials for root, called the machine or self host principal credentials, cannot be forwarded. To obtain DCE credentials that can be forwarded as a root user, a root user must issue **dce_login -f** with a DCE principal other than a self host principal.

Location

/usr/lpp/ssp/bin/pcp

Related Information

PSSP Commands: **dsh**, **hostlist**

AIX Commands: **rcp**

Examples

1. To copy a local file to host1 and host2 and rename it on those hosts, enter:

   ```
   pcp -w host1,host2 /etc/fileq /etc/filen
   ```

2. To copy a file in the current directory to a particular directory on all the nodes in the SP system, enter:

   ```
   pcp '-a' sysctl.acl /etc
   ```

3. To copy a directory subtree to all the hosts in the SP system that are currently responding, except for 'badnode,' enter:

   ```
   hostlist -av -e badnode | pcp -w - -r /etc/new /etc/new
   ```

4. To copy a directory subtree to all the hosts in the SP system that are currently responding, except for 'badnode,' enter:

```
pcp "-av -e badnode" -r /etc/new /etc/new
```

pdf

Purpose

pdf – Displays file system statistics on multiple nodes in parallel.

Syntax

pdf [−**w** − | *noderange* | **'***hostlist args***'**] [*file system1* [*file system2* ...]]

Flags

The **pdf** command requires the first flag or parameter on the command line to be a specification of the hosts on which the command is to be executed.

−**w** − Specifies that host names should be read from standard input. Host names can be in any format accepted by **rsh**.

Operands

noderange Indicates a specification via "node number." The node number corresponds to the position of a node in a frame and its slots. A node number indicates frame and slot. For example, frame 1 slot 1 would be referred to by 1. Frame 2 slot 1 would be node number 17, while frame 3 slot 2 would be 34. If a node occupies more than one slot, either node number refers to that node. Node numbers can be specified as ranges such as 1–3, which would refer to frame 1 slots 1–3, or 23–29,50,2, which would refer to frame 2 slots 7–13, frame 4 slot 2, and frame 1 slot 2. This option is only valid for an SP system.

'*hostlist args***'**
Specifies flags and arguments to be passed to the **hostlist** command. Hostlist allows several ways of listing hosts based on various criteria. Refer to the **hostlist** command.

Note: To use the working collective file specified by the WCOLL environment variable, you must specify a null string as the first argument. Refer to the **dsh** command for more information about a working collective.

Acceptable parameters to the **pdf** command are zero or more blank-separated file system names to be displayed. If no parameters are supplied, all local file systems (on each specified node) are displayed.

Description

The **pdf** command displays file systems and their usage statistics on one or more nodes in parallel. **pdf** is similar to the **df** command, but does not use it. Differences are:

- The **pdf** command can be executed on more than one node at a time.
- Execution is authorized to any authenticated user (this can be changed by the system administrator).
- The **pdf** command output contains more information than that of the **df** command.

Security

Since **pdf** uses **sysctl**, proper authentication and authorization to issue these commands is necessary.

Location

/usr/lpp/ssp/bin/pdf

Related Information

Commands: **hostlist**, **sysctl**

Examples

1. To list all file systems and their usage statistics on all hosts in the SP system, enter:

   ```
   pdf '-a'
   ```

2. To list the usage statistics for the **/tmp** file system on nodes named node1 and node2, enter:

   ```
   pdf -w node1,node2 /tmp
   ```

penotify

Purpose

penotify – Adds, removes, or shows AIX error log notification objects.

Syntax

penotify [–**a**] [–**c** *class*] [–**f add**] [–**l** *label*] [–**m** *method*] [–**n** *name*] [–**p** *pid*] [–**t** *type*] [–**w** *hosts*] [–**A** *alert*] [–**C** *rclass*] [–**N** *rname*] [–**P**] [–**T** *rtype*]

penotify [–**a**] [–**f remove**] [–**w** *hosts*] [–**n** *name*]

penotify [–**a**] [–**f show**] [–**w** *hosts*] [–**n** *name*]

Flags

–**a** Executes on all nodes in the system partition.

–**c** *class* Specifies the error class.

–**f** *func* Specifies the function: **add**, **remove**, or **show**.

–**h** Displays usage information.

–**l** *label* Specifies the error label.

–**m** *method* Specifies the notification method.

–**n** *name* Specifies the name of the notification object.

–**p** *pid* Specifies the process ID for the notification object.

–**t** *type* Specifies the error type.

–**w** *hosts* Runs the command on a file or a list of host names.

–**A** *alert* Specifies the match alertable errors (**true** or **false**).

–**C** *rclass* Specifies the resource class.

–**N** *rname* Specifies the resource name.

–**P** Specifies whether to persist across system restart (**yes**, if –**P** is provided).

–**T** *rtype* Specifies the resource type.

Operands

None.

Description

Use this command to add, remove, or show notification objects in the errnotify Object Data Management (ODM) class. The AIX **errdemon** matches logged errors to objects in this class to execute a method defined in the class object. The error class, error label, error type, alert, resource name, resource class, and resource type parameters are used for matching to logged errors. Refer to the *AIX Version 4 General Programming Concepts: Writing and Debugging Programs* for descriptions of error notification object class fields.

When a match occurs, the **errdemon** executes the notify method passing up to nine ($1–$9) parameters related to the logged error.

If the –**w** parameter begins with a slash (/), it is interpreted as a file containing a list of nodes to execute the command on; otherwise, it can be a comma-delimited list of host names or a single-quoted, blank-delimited list of host names. If neither the –**a** nor –**w** parameters are used, the command defaults to the local node.

Security

The **penotify** command consists of a client script and a server procedure which is executed by the Sysctl facility. Sysctl callbacks perform access authorization according to the configuration of security services on the server nodes. The server uses the Sysctl ACL callback for granting access which requires the caller to have a principal entry in the log management ACL: **/etc/logmgt.acl**. This principal must log into the proper authentication service prior to using this command.

Location

/usr/lpp/ssp/bin/penotify

Related Information

The *AIX Version 4 General Programming Concepts: Writing and Debugging Programs*

The *PSSP: Administration Guide*

Examples

1. To view all notification objects on nodes k4710, k4712, and k4715, enter:

   ```
   penotify -w k47n10,k47n12,k47n15 -f show
   ```

2. To remove the notification object named HDISK0_ERR on all nodes, enter:

   ```
   penotify -a -f remove -n HDISK0_ERR
   ```

3. To add a notification object to the nodes in the **/tmp/nodelist** file, enter:

   ```
   penotify -w /tmp/nodelist -f add -n PEND_ERR -P \

   -m'/spdata/sys1/EN_meth/EN_pend $1' -t PEND -c S
   ```

   This adds a notification object named PEND_ERR to all nodes in the **/tmp/nodelist** file. The object will **persist** when the system is restarted, and will match error records of type PEND and class S. The method that is executed by **errdemon** when a matching error occurs will be **/spdata/sys1/EN_meth/EN_pend**, and it will be passed the $1 parameter (sequence number). The notification method must be accessible to each node.

perspectives

Purpose

perspectives – Invokes the launch pad of the SP Perspectives graphical user interface (GUI).

Syntax

perspectives [–**userProfile** *name*] [–**systemProfile** *name*] [–**noProfile**] [–**backgroundColor** *colorName*] [–**foregroundColor** *colorName*] [–**fontFamily** *name*] [–**fontSize** *size*] [–**fontBold**] [–**fontItalic**] [–**nosplash**] [–**h**]

Flags

–**userProfile** *name*
: Upon initialization, loads the specified user profile. If a user profile named "Profile" exists in the user's home directory, it will be loaded by default if the –**userProfile** flag is not specified.

–**systemProfile** *name*
: Upon initialization, loads the specified system profile instead of the default system profile. The default system profile is named "Profile."

–**noProfile**
: Upon initialization, does not read either profile.

–**backgroundColor** *colorName*
: Overrides the background color specified by any profile or default with the specified color. Refer to Appendix A, "Perspectives Colors and Fonts" in *PSSP: Command and Technical Reference* for a list of valid color names.

–**foregroundColor** *colorName*
: Overrides any foreground color specified by any profile or default with the specified color. Refer to Appendix A, "Perspectives Colors and Fonts" in *PSSP: Command and Technical Reference* for a list of valid color names.

–**fontFamily** *name*
: Overrides any font family with the specified font. The list of valid family names is dependent on the X server. Refer to "Perspectives Fonts" in *PSSP: Command and Technical Reference* for a list of valid fonts.

–**fontSize** *size*
: Overrides any font point size with the specified size. Valid values are 6–30 points.

–**fontBold**
: Sets the font to bold.

–**fontItalic**
: Sets the font to italics.

–**nosplash**
: Does not display the splash screen before the Perspectives main window is displayed.

–**h**
: Displays usage information on the options available for the command.

Note: Most flags accepted by X will also be recognized. For example, –**display** *displayname*.

Operands

None.

Description

Use this command to invoke the SP Perspectives Launch Pad. The Launch Pad is a small, customizable GUI from which the user can start (or launch) executables associated with maintaining and monitoring an IBM RS/6000 SP.

The main window shows an icon for each executable that can be launched. Double-clicking on an icon launches the associated executable. To see descriptions of the applications that can be started from the Launch Pad select **Options ... Show Application Details**. To add, modify or delete applications select **Options ... Customize Applications**.

Preferences that define the look and layout of the Perspectives Launch Pad window are prioritized in the following order:

- Command line options
- User preferences profile
- System preferences profile
- Default values

Files

The User's Preferences are read from and saved to **$HOME/.perspectives(User Profile Name)**.

The System Preferences are read from and saved to **/usr/lpp/ssp/perspectives/profiles/$LANG/.perspectives(System Profile name)**.

The Perspectives application resource file **/usr/lpp/ssp/perspectives/app-defaults/$LANG/Perspectives**.

Security

Any user can run the **perspectives** command. However, only the icons that the user has AIX execute permission for will be shown in the main window.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

For information on using Perspectives, see the online help and the “Using the SP Perspectives” chapter in the *PSSP: Administration Guide*.

Location

/usr/lpp/ssp/bin/perspectives

Related Information

Specific Perspective windows may be brought up directly by invoking the following commands: **spevent**, **sphardware**, **spperfmon**, **spsyspar**, and **spvsd**.

Examples

1. To invoke the Perspectives Launch Pad, enter:

   ```
   perspectives
   ```

2. To force Perspectives to display a 14 point type regardless of what is set in the preference files, enter:

   ```
   perspectives -fontSize 14
   ```

pexec

Purpose

pexec – Specifies the parallel execution of a command.

Syntax

pexec [–**w** – | *noderange* | '*hostlist args*'] *command command_args*

Flags

The **pexec** commands requires the first flag or parameter on the command line to be a specification of the hosts on which the command is to be executed.

–**w** – Specifies that host names should be read from standard input. Host names can be in any format accepted by **rsh**.

Operands

noderange Indicates a specification via "node number." The node number corresponds to the position of a node in a frame and its slots. A node number indicates frame and slot. For example, frame 1 slot 1 would be referred to by 1. Frame 2 slot 1 would be node number 17, while frame 3 slot 2 would be 34. If a node occupies more than one slot, either node number refers to that node. Node numbers can be specified as ranges such as 1–3, which would refer to frame 1 slots 1–3, or 23–29,50,2, which would refer to frame 2 slots 7–13, frame 4 slot 2, and frame 1 slot 2. This option is only valid for an SP system.

'*hostlist args*'
Specifies flags and arguments to be passed to the **hostlist** command. Hostlist allows several ways of listing hosts based on various criteria. Refer to the **hostlist** command.

Note: To use the working collective file specified by the WCOLL environment variable, you must specify a null string as the first argument. Refer to the **dsh** command for more information about a working collective.

command Specifies a command to execute on the hosts in parallel.

cmd_args Specifies arguments to the command.

Description

The **pexec** command issues a command on multiple hosts in parallel. The output is formatted so that distinct output is displayed only once. The **pexec** command uses **dsh** to execute the specified command on multiple hosts. The output of the **ls** command is written to standard output and formatted. The **pls**, **prm**, **pmv**, **pfind**, and **pps** commands are simply links to **pexec**.

Note: If any of the **pls**, **prm**, **pfind**, **pps**, **pmv** are renamed, they do not work properly.

Files

working collective file
See the **dsh** command.

Security

You must have access to the AIX Secure Remote Commands to run this command.

This command will automatically forward the DCE credentials if K5 is an enabled AIX authentication method and the user of the command has DCE credentials that can be forwarded. The special DCE credentials for root, called the machine or self host principal credentials, cannot be forwarded. To obtain DCE credentials that can be forwarded as a root user, a root user must issue **dce_login -f**.

Location

/usr/lpp/ssp/bin/pexec

Related Information

Commands: **dsh**, **dshbak**, **hostlist**

Examples

1. To list the contents of **/usr** from each host1, host2, and host3 (described previously), enter:

```
pexec -w host1,host2,host3 ls /
```

2. To copy a directory subtree to all the hosts in the SP system that are currently responding, except for "badnode," enter:

```
hostlist -a -v -e badnode | pexec -w -cp -r /etc/new /etc/new
```

3. Another way to enter the command in the previous example follows:

```
pexec "-a -v -e badnode" cp -r /etc/new /etc/new
```

pexscr

Purpose

pexscr – Runs local and remote programs in parallel.

Syntax

pexscr

Flags

None.

Operands

None.

Description

The **pexscr** command executes particular commands on particular processors in parallel. **pexscr** reads lines of the following format from standard input:

```
host_name: arbitrary_command
```

and executes each *arbitrary_command* on the specified host. All commands are run in parallel. The AIX **rsh** command is used to run the remote commands, and local commands are run directly. Host names can include any parameter that may be specified on the **rsh** command.

Security

You must have access to the AIX Secure Remote Commands to run this command.

This command will automatically forward the DCE credentials if K5 is an enabled AIX authentication method and the user of the command has DCE credentials that can be forwarded. The special DCE credentials for root, called the machine or self host principal credentials, cannot be forwarded. To obtain DCE credentials that can be forwarded as a root user, a root user must issue **dce_login -f**.

Location

/usr/lpp/ssp/bin/pexscr

Related Information

Commands: **dsh**, **rsh**

Examples

To remove a file on host1 and rename a file on host2 simultaneously, enter:

```
pexscr <<END

host1: rm /tmp/shnozzola

host2: mv /tmp/shnozzola /tmp/bozo

END
```

pfck

Purpose

pfck – Displays file system statistics on multiple hosts in parallel based on usage criteria.

Syntax

pfck [–**w** – | *noderange* | '*hostlist args*'] { [–**s** *num*] [–**f** *num*] [–**u** *num*] [–**pf** *num*] [–**pu** *num*] [–**is** *num*] [–**if** *num*] [–**iu** *num*] [–**pif** *num*] [–**piu** *num*] }

Flags

The **pfck** command requires the first flag or parameter on the command line to be a specification of the hosts on which the command is to be executed.

–**w** – Specifies that host names should be read from standard input. Host names can be in any format accepted by **rsh**.

Subsequent flags specify file system usage statistics criteria to be applied in searching for file systems to display. At least one of the following flags must be specified. Multiple flags are allowed.

–**s** *num* Indicates that the file system size is > *num* kilobytes (KB).
–**f** *num* Indicates that the file system free space is < *num* kilobytes (KB).
–**u** *num* Indicates that the file system used space is > *num* kilobytes (KB).
–**pf** *num* Indicates that the file system free space is < *num* %.
–**pu** *num* Indicates that the file system used space is > *num* %.
–**is** *num* Indicates that the file system inodes are > *num*.
–**if** *num* Indicates that the file system free inodes are < *num*.
–**iu** *num* Indicates that the file system used inodes are > *num*.
–**pif** *num* Indicates that the file system free inodes are < *num* %.
–**piu** *num* Indicates that the file system used inodes are > *num* %.

File system usage statistics criteria are logically **OR**ed together when comparing against actual usage information. That is, if a file system meets *any* of the search criteria, then it is displayed.

Operands

noderange Indicates a specification via "node number." The node number corresponds to the position of a node in a frame and its slots. A node number indicates frame and slot. For example, frame 1 slot 1 would be referred to by 1. Frame 2 slot 1 would be node number 17, while frame 3 slot 2 would be 34. If a node occupies more than one slot, either node number refers to that node. Node numbers can be specified as ranges such as 1–3, which would refer to frame 1 slots 1–3, or 23–29,50,2, which would refer to frame 2 slots 7–13, frame 4 slot 2 and frame 1 slot 2. This option is only valid for an SP system.

'*hostlist args*'
Specifies flags and arguments to be passed to the **hostlist** command. Hostlist allows several ways of listing hosts based on various criteria. Refer to the **hostlist** command.

Note: To use the working collective file specified by the WCOLL environment variable, you must specify a null string as the first argument. Refer to the **dsh** command for more information about a working collective.

Description

The **pfck** command displays file systems and their usage statistics from one or more nodes in parallel based on usage criteria.

Since **pfck** uses **sysctl**, proper authentication and authorization to issue these commands is necessary.

Security

Since **pfck** uses **sysctl**, proper authentication and authorization to issue these commands is necessary.

Location

/usr/lpp/ssp/bin/pfck

Related Information

Commands: **hostlist**, **sysctl**

Examples

1. To list all file systems with less than 20% free space on all nodes in the SP system, enter:

   ```
   pfck -a -pf 20
   ```

2. To list all file systems on the nodes named node1, node2, and node4 which are greater than 98% full, enter:

   ```
   pfck -w node1,node2,node4 -pu 98
   ```

pfind

Purpose

pfind – Specifies a parallel find of files with a matching expression.

Syntax

| **pfind** [–**w** – | *noderange* | '*hostlist args*'] *find_args*

Flags

The **pfind** command requires the first flag or parameter on the command line to be a specification of the hosts on which the command is to be executed.

| –**w** – Specifies that host names should be read from standard input. Host
| names can be in any format accepted by **rsh**.

Operands

noderange Indicates a specification via "node number." The node number corresponds to the position of a node in a frame and its slots. A node number indicates frame and slot. For example, frame 1 slot 1 would be referred to by 1. Frame 2 slot 1 would be node number 17, while frame 3 slot 2 would be 34. If a node occupies more than one slot, either node number refers to that node. Node numbers can be specified as ranges such as 1–3, which would refer to frame 1 slots 1–3, or 23–29,50,2, which would refer to frame 2 slots 7–13, frame 4 slot 2, and frame 1 slot 2. This option is only valid for an SP system.

'*hostlist args*'
Specifies flags and arguments to be passed to the **hostlist** command. Hostlist allows several ways of listing hosts based on various criteria. Refer to the **hostlist** command.

Note: To use the working collective file specified by the WCOLL environment variable, you must specify a null string as the first argument. Refer to the **dsh** command for more information about a working collective.

find_args Specifies arguments to the AIX **find** command.

Description

The **pfind** command issues the AIX **find** command on multiple hosts. The output is formatted so that distinct output is displayed only once. The **pfind** command uses **dsh** to execute the **find** command on multiple hosts. The output of the **ls** commands is written to standard output and formatted. The **pfind** command is identical to **pexec find**.

Files

working collective file
See the **dsh** command.

Security

You must have access to the AIX Secure Remote Commands to run this command.

This command will automatically forward the DCE credentials if K5 is an enabled AIX authentication method and the user of the command has DCE credentials that can be forwarded. The special DCE credentials for root, called the machine or self host principal credentials, cannot be forwarded. To obtain DCE credentials that can be forwarded as a root user, a root user must issue **dce_login -f**.

Location

/usr/lpp/ssp/bin/pfind

Related Information

Commands: **dsh**, **find**, **hostlist**, **pexec**

Examples

To find out if the file **elvis** is contained in **/usr/bin** on any host1, host2, and host3 (described previously), enter:

```
pfind -w host1,host2,host3 /usr/bin -print -name "elvis"
```

pfps

Purpose

pfps – Finds and performs operations on processes on multiple hosts in parallel based on the value of an expression.

Syntax

| **pfps** [–**w** – | *noderange* | '*hostlist args*'] *operation* [*expression*]

Flags

The **pfps** command requires the first flag or parameter on the command line to be a specification of the hosts on which the command is to be executed.

| –**w** – Specifies that host names should be read from standard input. Host
| names can be in any format accepted by **rsh**.

Operands

noderange Indicates a specification via "node number." The node number corresponds to the position of a node in a frame and its slots. A node number indicates frame and slot. For example, frame 1 slot 1 would be referred to by 1. Frame 2 slot 1 would be node number 17, while frame 3 slot 2 would be 34. If a node occupies more than one slot, either node number refers to that node. Node numbers can be specified as ranges such as 1–3, which would refer to frame 1 slots 1–3, or 23–29,50,2, which would refer to frame 2 slots 7–13, frame 4 slot 2, and frame 1 slot 2. This option is only valid for an SP system.

'*hostlist args*'
Specifies flags and arguments to be passed to the **hostlist** command. Hostlist allows several ways of listing hosts based on various criteria. Refer to the **hostlist** command.

Note: To use the working collective file specified by the WCOLL environment variable, you must specify a null string as the first argument. Refer to the **dsh** command for more information about a working collective.

operation Must be at least one of the following:

–**print** If expression is not specified, all process information is listed. If expression is specified, process information for which the expression is true is listed. Output is in **ps auw** format.

–**kill** *signal* Causes matching processes to be stopped with the specified signal if the user is authorized or owns the process. The signal can be specified as a number or the name of the signal (for example, HUP).

–**nice** *n* Causes the nice value to be set for matching processes if the user is authorized.

expression The *expression* must be at least one of the following:

–n *name*
: Evaluates to true if the fully qualified name of the process matches the name.

–tn *name*
: Evaluates to true if the name of the process matches the name.

–o *owner*
: Evaluates to true if the user name of the process matches the owner.

–pty *name*
: Evaluates to true if the name matches the process controlling terminal.

–rtime *hh:mm*
: Evaluates to true for processes whose total execution time is *hh:mm* time or longer.

–stime *dd:hh:mm*
: Evaluates to true for processes that have started at least *dd* days, *hh* hours, and *mm* minutes ago.

–r *state*
: Evaluates to true for processes in the specified run state.

–cpu *percentage*
: Evaluates to true for processes using greater than the specified percentage of system CPU.

–mem *percentage*
: Evaluates to true for processes consuming more than the specified percentage of system memory.

–or
: Evaluates the expression by **OR**ing together the terms.

Description

The **pfps** command performs operations on processes on one or more hosts in parallel. These operations include printing information about processes (–**print**), sending a signal to the process (–**kill**), and changing the priority of the process (–**nice**).

Authorization is via an Access Control List (ACL) on each node and is required when users try to **kill** a process that they do not own or **nice** a process to a higher priority. ACLs for **pfs** are contained in the **/etc/sysctl.pfps.acl** file.

An expression can also be specified using the preceding flags to select processes for when the expression evaluates to true. Flags are **AND**ed together unless the –**or** flag is used.

Parentheses can be used to group flags, but parenthesis must be separated from flags by a space. Also, parenthesis or any special shell character should be escaped with a backslash (\).

Since **pfps** uses **sysctl**, proper authentication and authorization to issue these commands is necessary.

Files

/etc/sysctl.pfps.acl
The ACL file which authorizes listed principals to use the **nice** and **kill** options.

Security

Since **pfps** uses **sysctl**, proper authentication and authorization to issue these commands is necessary.

Location

/usr/lpp/ssp/bin/pfps

Related Information

Commands: **hostlist**, **kill**, **nice**, **ps**, **renice**, **sysctl**

Examples

1. To list all processes on all hosts in the SP system, enter:

   ```
   pfps '-a' -print
   ```

2. To restart **daemon** processes on host1 and host2 that were running for more than one day (the user must be listed in the **/etc/sysctl.pfps.acl** on each host or the command is ignored for that host), enter:

   ```
   pfps -w host1,host2 -rtime 24:00 -tn daemond -kill HUP
   ```

3. To list all processes belonging to root that are using more than 10% of system CPU or 10% of system memory on hosts listed in the **./wcollective** file, enter:

   ```
   WCOLL=./wcollective pfps '' \( -cpu 10 -or -mem 10 \) -o root -print
   ```

pls

Purpose

pls – Specifies a parallel list of files and directories.

Syntax

pls [**–w –** | *noderange* | **'***hostlist args***'**] *ls_args*

Flags

The **pls** command requires the first flag or parameter on the command line to be a specification of the hosts on which the command is to be executed.

–w – Specifies that host names should be read from standard input. Host names can be in any format accepted by **rsh**.

Operands

noderange Indicates a specification via "node number." The node number corresponds to the position of a node in a frame and its slots. A node number indicates frame and slot. For example, frame 1 slot 1 would be referred to by 1. Frame 2 slot 1 would be node number 17, while frame 3 slot 2 would be 34. If a node occupies more than one slot, either node number refers to that node. Node numbers can be specified as ranges such as 1–3, which would refer to frame 1 slots 1–3, or 23–29,50,2, which would refer to frame 2 slots 7–13, frame 4 slot 2, and frame 1 slot 2. This option is only valid for an SP system.

'*hostlist args***'**
Specifies flags and arguments to be passed to the **hostlist** command. Hostlist allows several ways of listing hosts based on various criteria. Refer to the **hostlist** command.

Note: To use the working collective file specified by the WCOLL environment variable, you must specify a null string as the first argument. Refer to the **dsh** command for more information about a working collective.

ls_args Specifies arguments to the AIX **ls** command.

Description

The **pls** command issues the AIX **ls** command on multiple hosts. The output is formatted so that duplicate output is displayed only once. The **pls** command uses **dsh** to execute the **ls** command on multiple hosts. The output of the **ls** commands is written to standard output and formatted. The **pls** command is identical to **pexec ls**.

Files

working collective file
See the **dsh** command.

Security

You must have access to the AIX Secure Remote Commands to run this command.

This command will automatically forward the DCE credentials if K5 is an enabled AIX authentication method and the user of the command has DCE credentials that can be forwarded. The special DCE credentials for root, called the machine or self host principal credentials, cannot be forwarded. To obtain DCE credentials that can be forwarded as a root user, a root user must issue **dce_login -f**.

Location

/usr/lpp/ssp/bin/pls

Related Information

Commands: **dsh**, **ls**, **pexec**

Examples

To list the contents of **/usr** from each host1, host2, and host3 (described previously), enter:

```
pls -w host1,host2,host3 /usr
```

pmanchown

Purpose

pmanchown – Changes the ownership of a user's Problem Management subscriptions.

Syntax

pmanchown
{–**d** *Old_DCE_Principal* | –**k** *Old_Kerberos_V4_Principal* | {–**u** *Old_AIX_Owner* –**n** *Old_AIX_Hostname*}}
{–**D** *New_DCE_Principal* | –**K** *New_Kerberos_V4_Principal* | {–**U** *New_AIX_Owner* –**N** *New_AIX_Hostname*}}

Flags

–**d** Specifies the old DCE principal name.

–**k** Specifies the old Kerberos Version 4 principal name.

–**u** Specifies the old AIX owner name.

–**n** Specifies the old AIX hostname.

–**D** Specifies the new DCE principal name.

–**K** Specifies the new Kerberos Version 4 principal name.

–**U** Specifies the new AIX owner name.

–**N** Specifies the new AIX hostname.

Operands

None.

Description

Use the **pmanchown** command to change the ownership of a user's Problem Management subscriptions. The first set of arguments (where the flags are all lowercase letters) identify the old subscription owner. The second set of arguments (where the flags are all uppercase letters) identify the new subscription owner. Note that the –**d**, –**k**, and –**u** flags are mutually exclusive; you cannot identify the old subscription owner by multiple ownership attributes. The –**D**, –**K**, and –**U** flags are also mutually exclusive; you cannot identify the new subscription owner by multiple ownership attributes.

The **pmanchown** command simply finds all subscriptions belonging to the old owner and transfers the ownership to the new owner. After running the command, the system administrator should refresh all of the **pmand** daemons in the system partition. If the **pmanchown** command is run several times in succession, then you can delay refreshing the **pmand** daemons until after the last invocation of **pmanchown**.

Security

You must have write access to the SDR to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Location

/usr/lpp/ssp/bin/pmanchown

Examples

To transfer ownership of all subscriptons which belong to the Kerveros Version 4 principal joe@xyz.com to the DCE principal /.../xyz.com/joe, enter:

```
pmanchown -k joe@XYZ.com -D /.../xyz.com/joe
```

pmanctrl

Purpose

pmanctrl – Controls the Problem Management subsystem.

Syntax

pmanctrl {–**a** | –**s** | –**k** | –**d** | –**c** | –**t** | –**o** | –**r** | –**h**}

Flags

–**a** Adds the Problem Management subsystem.

–**s** Starts the Problem Management subsystem.

–**k** Stops the Problem Management subsystem.

–**d** Deletes the Problem Management subsystem.

–**c** Deletes the Problem Management subsystem from all system partitions.

–**t** Turns tracing on for the Problem Management subsystem.

–**o** Turns tracing off for the Problem Management subsystem.

–**r** Required by the **syspar_ctrl** command, but does not do anything.

–**h** Displays usage information.

Operands

None.

Description

Problem Management is a general purpose facility for monitoring and reacting to specific event occurrences within the SP system. The **pmanctrl** command controls the operations of the subsystems that are required for Problem Management. These subsystems are under the control of the System Resource Controller (SRC) and belong to a subsystem group called **pman**. Associated with each subsystem is a daemon.

An instance of the Problem Management subsystem executes on the control workstation and on every node of a system partition. Because Problem Management provides its services within the scope of a system partition, its subsystems are said to be system partition sensitive. For this reason, the **pmanctrl** command is normally invoked by the **syspar_ctrl** command during installation of the system, boot or reboot of individual nodes, and partitioning or repartitioning of the system.

From an operational point of view, the Problem Management subsystem group is organized as follows:

Subsystem: Problem Management

Subsystem Group: pman

SRC subsystems: pman and **pmanrm**

The **pman** subsystem is associated with the **pmand** daemon. The **pmanrm** subsystem is associated with the **pmanrmd** daemon.

The subsystem names on the nodes are **pman** and **pmanrm**. There is one of each subsystem per node and it is associated with the system partition to which the node belongs.

On the control workstation, there are multiple instances of each subsystem, one for each system partition. Accordingly, the subsystem names on the control workstation have the system partition name appended to them. For example, for system partitions named **sp_prod** and **sp_test**, the subsystems on the control workstation are named **pman.sp_prod**, **pman.sp_test**, **pmanrm.sp_prod**, and **pmanrm.sp_test**.

Daemons: **pmand** and **pmanrmd**

The **pmand** daemon provides the majority of Problem Management functions.

The **pmanrmd** daemon provides command-based resource monitor data to the **pmand** daemon.

The **pmanctrl** command provides a variety of controls for operating the Problem Management subsystems:

- Adding, starting, stopping, and deleting the subsystems
- Cleaning up the subsystems, that is, deleting them from all system partitions
- Turning tracing on and off

Unless the **−c** flag is used, the **pmanctrl** command only operates within a single partition. On a node, the **pmanctrl** command operates within the system partition to which the node belongs. On the control workstation, the **pmanctrl** command operates within any single partition, which can be chosen by setting the SP_NAME environment variable.

When the **pmanctrl** command is called with the **−a** flag, it uses the **mkssys** command to add the subsystems to the SRC, and it takes the necessary steps to make sure that the subsystems are automatically started when the node is booted.

When the **pmanctrl** command is called with the **−s** flag, it uses the **startsrc** command to start the **pman** and **pmanrm** subsystems.

When the **pmanctrl** command is called with the **−k** flag, it uses the **stopsrc** command to stop the **pman** and **pmanrm** subsystems.

When the **pmanctrl** command is called with the **−d** flag, it uses the **rmssys** command to delete the subsystems from the SRC, and if there are no more Problem Management subsystems remaining, it makes sure there is no **/etc/inittab** entry for the Problem Management subsystem.

When the **pmanctrl** command is called with the **−c** flag, it stops all running Problem Management subsystems, removes them all from the SRC, and makes sure there is no **/etc/inittab** entry for the Problem Management subsystem.

When the **pmanctrl** command is called with the **−t** flag, it uses the **traceson** command to turn on tracing in the **pman** subsystem. Tracing is not available for the **pmanrmd** subsystem.

When the **pmanctrl** command is called with the **−o** flag, it uses the **tracesoff** command to turn off tracing in the **pman** subsystem. Tracing is not available for the **pmanrmd** subsystem.

While they are running, the Problem Management daemons provide information about their operation and errors by writing entries in a log file that is located in the **/var/adm/SPlogs/pman** directory. On the control workstation, the **pmand** daemon writes to a log file named **pmand.***syspar_name***.log**, and the **pmanrmd** daemon writes to a log file named **pmanrmd.***syspar_name***.log**, where *syspar_name* is the name of the system partition. On the nodes, the **pmand** daemon writes to a log file named **pmand.log** and the **pmanrmd** daemon writes to a log file named **pmanrmd.log**.

Exit Values

0 Indicates the successful completion of the command.

nonzero Indicates that an error occurred.

Security

You must have root privilege to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

The "Using the Problem Management Subsystem" chapter in *PSSP: Administration Guide*

IBM AIX Version 4 Commands Reference

Information about the System Resource Controller (SRC) in *IBM AIX Version 4 General Programming Concepts: Writing and Debugging Programs*

Location

/usr/lpp/ssp/bin/pmanctrl

Related Information

SP Command: **syspar_ctrl**

AIX Commands: **mkssys**, **rmssys**, **startsrc**, **stopsrc**

Examples

1. To add the Problem Management subsystem to the SRC, enter:

   ```
   pmanctrl -a
   ```

2. To start the Problem Management subsystem, enter:

   ```
   pmanctrl -s
   ```

3. To stop the Problem Management subsystem, enter:

   ```
   pmanctrl -k
   ```

4. To delete the Problem Management subsystem from the SRC, enter:

   ```
   pmanctrl -d
   ```

5. To clean up the Problem Management subsystem on all system partitions, enter:

   ```
   pmanctrl -c
   ```

6. To turn tracing on for the Problem Management daemon, enter:

   ```
   pmanctrl -t
   ```

7. To turn tracing off for the Problem Management daemon, enter:

   ```
   pmanctrl -o
   ```

pmandef

Purpose

pmandef – Defines events and resulting actions to the Problem Management subsystem.

Syntax

To subscribe to an event and associate a set of actions with that event, use the following:

pmandef –**s** *HandleName* –**e** *ResourceVariable:ResourceIdentifier:Expression*
[–**r** *RearmExpression*] [–**i**] [–**c** *EventCommand*] [–**C** *RearmCommand*]
[–**t** *EventTrapid*] [–**T** *RearmTrapid*] [–**l** *EventLogText*]
[–**L** *RearmLogText*] [–**x** *EventCmdTimeout*]
[–**X** *RearmCmdTimeout*] [–**U** *UserName*] [–**m** *UserLabel*]
[–**h** {*Host1*[,*Host2*,...] | – | **local**} | –**N** *NodeGroup* | –**n** *NodeRange*]

To deactivate or activate a Problem Management subscription, use the following:

pmandef {–**d** | –**a**} *HandleName* [–**h** {*Host1*[, *Host2*,...] | –} |
–**N** *NodeGroup* | –**n** *NodeRange*]
or
pmandef {–**d** | –**a**} **all**

To query or remove a Problem Management subscription, use the following:

pmandef {–**q** | –**u**} { *HandleName* | **all**}

Flags

Flags that specify the type of request follow:

–**s** *HandleName*
: Specifies that this is a subscribe request and the remaining flags define the Problem Management subscription. The *HandleName* provides a means to identify this subscription to Problem Management using the –**d**, –**a**, –**u**, or –**q** flags. The **all** keyword cannot be used as a handle name for a subscribe request.

–**d {***HandleName* **| all}**
: Specifies that the actions associated with the subscription identified by *HandleName* should be turned off or deactivated. The **all** keyword deactivates all subscriptions owned by the user . The –**h**, –**N**, and –**n** flags cannot be used with the **all** keyword on a deactivate request.

–**a {***HandleName* **| all}**
: Specifies that the actions associated with the subscription identified by *HandleName* should be turned on or activated. The **all** keyword activates all subscriptions owned by the user . The –**h**, –**N**, and –**n** flags cannot be used with the **all** keyword on an activate request.

–**u {***HandleName* **| all}**
: Specifies that this is an unsubscribe request and the subscription identified by *HandleName* should be removed. The **all** keyword unsubscribes all subscriptions owned by the user .

−q {*HandleName* **| all}**
Requests all of the Problem Management daemons, for which the subscription identified by *HandleName* is defined, to provide status about the named subscription. The **all** keyword queries all subscriptions owned by the user .

Flags that specify the hosts to be affected by the request follow:

−h {*Host1***[,***Host2***,...] | − | local}**
For a subscribe request, specifies the hosts that belong to the subscription. For an activate or deactivate request, specifies the hosts to be activated or deactivated, which can only include hosts that belong to the subscription. The hosts may be specified as a comma-separated list of host names or the hyphen (-) may be used to indicate that host names are to be read from standard input, or the **local** keyword may be used to indicate that the list of hosts is to be obtained from the *NodeNum* field in the event instance vector. Use of the **local** keyword also causes any resulting actions to occur on the same host where the event occurs. All specified hosts must reside in the same system partition. The **local** keyword is not allowed for an activate or deactivate request.

−N *NodeGroup*
For a subscribe request, specifies the node group that contains all of the hosts that belong to the subscription. For an activate or deactivate request, specifies the node group that contains all of the hosts to be activated or deactivated, which can only include hosts that belong to the subscription. All specified hosts must be in a partitioned-bound node group.

−n *NodeRange*
For a subscribe request, specifies the node range that contains all of the hosts that belong to the subscription. For an activate or deactivate request, specifies the node range that contains all of the hosts to be activated or deactivated, which can only include hosts that belong to the subscription. A node range is a series of numbers that are separated by commas and hyphens, which indicate a set of node numbers, such as 0−3,5,8−10. All specified hosts must reside in the same system partition.

Flags that define a Problem Management subscription follow:

−e *ResourceVariable***:***ResourceIdentifier***:***Expression*
Specifies the Event Management resource variable, resource identifier and expression that define the event for which actions are generated for this Problem Management subscription. Refer to *IBM RS/6000 Cluster Technology: Event Management Programming Guide and Reference* for further information about Event Management event definitions.

−r *RearmExpression*
Specifies the Event Management re-arm expression, which together with the resource variable, resource identifier and expression specified by the **−e** flag, defines the re-arm event for which actions are generated for this Problem Management subscription. Refer to *IBM RS/6000 Cluster Technology: Event Management Programming Guide*

and Reference for further information about Event Management event definitions.

–**i**
Specifies that the Problem Management subsystem should not request an immediate evaluation of the event defined by the **-e** flag, when the event is registered with the Event Management subsystem. After it receives the event registration, Event Management must wait until the next observation of the resource before it may generate events for the event registration that belongs to this subscription. If you do not specify the **-i** flag, Event Management may generate an event based on the current value of the resource, even if the resource value has not changed recently. This flag is ignored by the Problem Management subsystem on hosts running PSSP releases older than PSSP 3.1.

–**c** *EventCommand*
Specifies a command to be executed when the event defined by the –**e** flag occurs. The command will be interpreted by the user's login program, so *EventCommand* may contain additional arguments and shell metacharacters. For example:

```
echo this is a test >/tmp/event.out
```

is allowed.

–**C** *RearmCommand*
Specifies a command to be executed when the re-arm event defined by the –r flag occurs. The command will be interpreted by the user's login program, so *RearmCommand* may contain additional arguments and shell metacharacters. For example:

```
echo this is a test >/tmp/rearm.out
```

is allowed.

–**t** *EventTrapid*
Specifies that a Simple Network Management Protocol (SNMP) trap should be sent when the event defined by the –**e** flag occurs. *EventTrapid* is the specific trap ID to be used.

–**T** *RearmTrapid*
Specifies that an SNMP trap should be sent when the re-arm event defined by the –**r** flag occurs. *RearmTrapid* is the specific trap ID to be used.

–**l** *EventLogText*
Specifies text that should be written to the AIX error log and BSD syslog when the event defined by the –**e** flag occurs.

–**L** *RearmLogText*
Specifies text that should be written to the AIX error log and BSD syslog when the re-arm event defined by the –**r** flag occurs.

–**x** *EventCmdTimeout*
Specifies a time limit in seconds for the command specified by the –**c** flag. If the command does not complete within *EventCmdTimeout* seconds, it will be sent a SIGTERM signal. If the command does not terminate after an additional 5 seconds, the SIGTERM signal will be followed by a SIGKILL signal. If the –**x** flag is not specified, the command runs until completion.

−**X** *RearmCmdTimeout*
Specifies a time limit in seconds for the command specified by the −**C** flag. If the command does not complete within *RearmCmdTimeout* seconds, it will be sent a SIGTERM signal. If the command does not terminate after an additional 5 seconds, the SIGTERM signal will be followed by a SIGKILL signal. If the −**X** flag is not specified, the command runs until completion.

−**U** *UserName*
Specifies the user to run the commands specified by the −**c** and −**C** flags. If the −**U** flag is omitted, the commands will run as the user who issued the subscribe request.

−**m** *UserLabel*
Specifies a tag to associate with the subscription. The Problem Management subsystem does not use this tag for anything, so the user can specify anything as a tag. The tag can be retrieved with the **pmanquery** command.

Operands

None.

Description

The Problem Management subsystem allows you to specify that an action takes place as the result of a specific event occurrence within the SP system. The Problem Management subsystem registers for the event with the Event Management subsystem. When the Event Management subsystem reports the occurrence of the event back to Problem Management, the Problem Management subsystem performs the requested action on your behalf. The actions that are performed as the result of an event can be any or all of the following:

- A command is executed
- An entry is written to the AIX error log and BSD syslog
- An SNMP trap is generated

The **pmandef** command is the command line interface for making such requests to the Problem Management subsystem. For example, running the following command on node 5:

```
pmandef -s Program_Monitor \

-e 'IBM.PSSP.Prog.pcount:NodeNum=12;ProgName=mycmd;UserName=bob:X@0==0' \

-r "X@0==1" -c "echo program has stopped >/tmp/myevent.out" \

-C "echo program has restarted >/tmp/myrearm.out"
```

causes the command `echo program has stopped >/tmp/myevent.out` to run on node 5 whenever the number of processes named “mycmd” and owned by user “bob” on node 12 becomes 0. When this number increases back to 1, the command `echo program has restarted >/tmp/myrearm.out` runs on node 5.

If you do not want the command to run on the same node from which the **pmandef** command was issued, then use one of the −**h**, −**N**, or −**n** flags.

The following example causes the commands to run on both k21n01 and k21n02 whenever bob's program dies or gets restarted on either nodes 12 or 13.

```
pmandef -s Program_Monitor \

-e 'IBM.PSSP.Prog.pcount:NodeNum=12-13;ProgName=mycmd;UserName=bob:X@0==0' \

-r "X@0==1" -c /usr/local/bin/start_recovery \

-C /usr/local/bin/stop_recovery -h k21n01,k21n02
```

The following example causes the commands to run on nodes 1, 2, 3, and 7 whenever bob's program dies or gets restarted on any of nodes 1, 2, 3, 4, 5, or 13. If bob's program dies on node 4, the command **/usr/local/bin/start_recovery** runs on nodes 1, 2, 3, and 7.

```
pmandef -s Program_Monitor \

-e 'IBM.PSSP.Prog.pcount:NodeNum=1-5,13;ProgName=mycmd;UserName=bob:X@0==0' \

-r "X@0==1" -c /usr/local/bin/start_recovery \

-C /usr/local/bin/stop_recovery -n 1-3,7
```

If you want to define a subscription for more than one node but you want the command to run only on the same node where the event occurs, use the −**h local** option. Consider the following command:

```
pmandef -s Filesystem_Monitor \

-e 'IBM.PSSP.aixos.FS.%totused:NodeNum=10-14;VG=myvg;LV=mylv:X>95' \

-l "filesystem is almost full" -h local
```

Whenever the file system associated with the “ mylv” logical volume and “ myvg” volume group on node 11 becomes more than 95 percent full, the text `filesystem is almost full` gets written to the AIX error log and BSD syslog only on node 11. Whenever the same thing occurs on node 12, the same text gets written to the AIX error log and BSD syslog only on node 12. The file system filling up on node 11 is really a separate event than the file system filling up on node 12 or node 13, and the −**h local** option is just a convenient way to define actions for several events at the same time.

Issuing the **pmandef** command with the −**s** flag to associate an action with an event creates a Problem Management “subscription.” When you issue the **pmandef** command to create a Problem Management subscription, the definition of the subscription gets stored in the System Data Repository (SDR) so the definition becomes permanent. As soon as the subscription gets stored in the SDR, the **pmandef** command also requests the affected Problem Management daemons within the SP system to start acting on the new subscription. Since it is possible for some of the nodes affected by this to be down, the **pmandef** command is considered successful once the subscription is stored in the SDR. The inability to reach all of the affected Problem Management daemons is not considered to be an error, because they will eventually pick up the new subscription once they get restarted.

If the Event Management resource variable, resource identifier, expression, or re-arm expression contains an error, the error will not get discovered until the

Problem Management daemon registers for the event with the Event Management subsystem. When this happens, the subscription definition does not automatically get removed from the SDR. You must remove the subscription by using the **pmandef** command with the −**u** flag. The argument to the −**u** flag is the same name that was previously specified as the argument to the −**s** flag.

The **pmandef** command with the −**u** flag removes the subscription definition from the SDR and tells the appropriate Problem Management daemons to stop monitoring for the event associated with that subscription. The **pmandef** command with the −**d** can be used to turn off or “deactivate” a subscription. This also tells the appropriate Problem Management daemons to stop monitoring for the event associated with that subscription, but it does not remove the subscription definition from the SDR. The subscription remains deactivated until you call **pmandef** with the −**a** flag to “activate” the subscription. You can also use the **pmandef** command with the −**q** flag to query the appropriate Problem Management daemons for status about a subscription. This just tells you whether each daemon is currently monitoring for the event associated with that subscription.

The **pmandef** command requires the user to have Problem Management access on the local node to make the necessary changes to the SDR. If the user does not have Problem Management access on the local node, the command will be unsuccessful. The user should also have Problem Management access on the nodes that are affected by the subscription for the Problem Management daemons on those nodes to dynamically process the request. If the user does not have Problem Management access to any of these nodes, the Problem Management daemons on these nodes will not process the request until the next time they are restarted or refreshed. See the "Using the Problem Management Subsystem" chapter in the PSSP: Administration Guide for information on obtaining Problem Management access in each of the supported PSSP trusted services security configurations.

When a new subscription is created using the **pmandef** −**s** command, the user's current security context is used to establish ownership of the subscription. Modifications to the subscription using the **pmandef** command with the −**u**, −**d** and −**a** flags are only allowed by a user who can be identified as the owner of the subscription. See the "Using the Problem Management Subsystem" chapter in the PSSP: Administration Guide for a complete description of subscription ownership in all of the supported PSSP trusted services security configurations. Subscription ownership is also used to decide whether the action that results from an event should be allowed.

After the Problem Management daemon receives notification that an event has occurred, and before it performs the action for that event, the Problem Management daemon checks to see whether the subscription owner is authorized to perform the requested action on the node where the Problem Management daemon is running. If the requested action is an entry in the AIX error log and BSD syslog or the generation of an SNMP trap, the subscription owner must have AIX Secure Remote Commands access to the node as the root user. If the requested action is execution of a command, the subscription owner must have AIX Secure Remote Commands access to the node as the command user. The command user is by default the same user who issued the **pmandef** −**s** command to create the subscription. A different user can be specified to **pmandef** by using the −**U** flag. See the "Using the Problem Management Subsystem" chapter in the PSSP:

Administration Guide for a complete description of obtaining AIX Secure Remote Commands access.

Exit Values

0 Indicates the successful completion of the command.

1 Indicates the successful completion of the command, but the output contains a warning message.

2 Indicates that the command was unsuccessful.

Security

You must have Problem Management access to run this command.

You must have the appropriate AIX Secure Remote Commands access for requested actions to be executed in response to events.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

IBM RS/6000 Cluster Technology: Event Management Programming Guide and Reference

The “Using the Problem Management Subsystem” chapter in *PSSP: Administration Guide*

Location

/usr/lpp/ssp/bin/pmandef

Related Information

Commands: **pmanquery**, **sysctl**

Examples

In this example, a user issues the command:

```
pmandef -s Program_Monitor \

-e 'IBM.PSSP.Prog.pcount:NodeNum=12;ProgName=mycmd;UserName=relish:X@0==0' \

-r "X@0==1" -c /usr/local/bin/start_recovery \

-C /usr/local/bin/stop_recovery -n 5 -U ketchup
```

A subscription named “Program_Monitor” will be created on the local node if the user has Problem Management access. The requested actions for this subscription are for node 5, so the user must also have Problem Management access on node 5, in order for the Problem Management daemon on node 5 to act on this subscription immediately. Otherwise, the Problem Management daemon will not act on this subscription until the next time it gets restarted.

This subscription requests that whenever the number of processes named “mycmd” and owned by user relish on node 12 becomes 0, the command start_recovery will run on node 5. When the number of processes increases back to 1, the command stop_recovery will run on node 5. The subscription also requests that the commands start_recovery and stop_recovery are to run on node 5 as the user ketchup.

Once the subscription is created, any user who can be identified as the subscription owner can deactivate, activate, or remove this subscription.

To deactivate, enter:

```
pmandef -d Program_Monitor
```

To activate, enter:

```
pmandef -a Program_Monitor
```

To remove, enter:

```
pmandef -u Program_Monitor
```

pmanquery

Purpose

pmanquery – Queries the System Data Repository (SDR) for a description of one or more Problem Management subscriptions.

Syntax

pmanquery –**n** {*HandleName* | **all**} [–**q**] [–**k** {*KerberosV4Principal* | **all**}] [–**p** {*DCEPrincipal* | **all**}] [–**U** {*AIXUsername* | **all**} –**H** {*Hostname* | **all**}] [–**a** | –**d** | –**t** | –**x**]

Flags

Flags that specify the scope of the search follow:

–**n {***HandleName* **| all}**
: Searches for Problem Management subscriptions with the specified handle name that was previously given as the argument to the –**s** flag of the **pmandef** command. The **all** keyword allows any handle name to be selected by the search of the SDR. The **all** keyword cannot be used in conjunction with the –**a**, –**d**, or –**t** flags.

–**k {***KerberosV4Principal* **| all}**
: Searches for Problem Management subscriptions owned by the specified Kerberos Version 4 principal. The **all** keyword allows any Kerberos Version 4 principal to be selected by the search of the SDR. The **all** keyword cannot be used in conjunction with the –**a**, –**d**, or –**t** flags.

–**p {***DCEPrincipal* **| all}**
: Lists Problem Management subscriptions owned by the specified DCE principal. The **all** keyword allows any DCE principal to be selected by the search of the SDR. The **all** keyword cannot be used in conjunction with the –**a**, –**d**, or –**t** flags.

–**U {***UserName* **| all}**
: Lists Problem Management subscriptions owned by the specified AIX username. The **all** keyword allows any AIX user name to be selected by the search of the SDR. The **all** keyword cannot be used in conjunction with the –**a**, –**d**, or –**t** flags. The –**U** and –**H** flags may only be used together to specify an AIX username and hostname combination.

–**H {***Hostname* **| all}**
: Lists Problem Management subscriptions owned by the specified hostname. The **all** keyword allows any hostname to be selected by the search of the SDR. The **all** keyword cannot be used in conjunction with the –**a**, –**d**, or –**t** flags. The –**U** and –**H** flags may only be used together to specify an AIX username and hostname combination.

Flags that format the output follow:

–**q**
: Suppresses the output of the header line.

–**a**
: Lists output by node numbers in column format that represent the set of nodes that have not "deactivated" the subscription identified by the –**n** flag. The **all** keyword cannot be used with the –**n**, –**k**, –**p**, –**U** or –**H** flags when –**a** is used.

−d Lists output by node numbers in column format that represent the set of nodes that have “deactivated” the subscription identified by the **−n** flag. The **all** keyword cannot be used with the **−n**, **−k**, **−p**, **−U** or **−H** flags when **−d** is used.

−t Lists output by node numbers in column format that represent the set of all nodes that recognize the subscription identified by the **−n** flag. This is the sum of the node lists that get printed by the **−a** and **−d** flags. The **all** keyword cannot be used with the **−n**, **−k**, **−p**, **−U** or **−H** flags when **−t** is used.

−x Adds three additional fields to the output: **pmTotalNodeList**, **pmActivatedList** and **pmDeactivatedList**. These fields each represent a set of node numbers as a series of hexadecimal digits which form a bitmask, where the rightmost bit position represents node 0, and the node numbers increase as you shift to the left. If the bit position for a node number is "on", the node number belongs to the set. Otherwise, the node number does not belong to the set. For example, the bitmask "05" represents the set of nodes 0 and 2. The bitmask "1c93" represents the set of nodes 0, 1, 4, 7, 10, 11 and 12. The **pmTotalNodeList**, **pmActivatedList** and **pmDeactivatedList** fields use this format to represent the same sets of node numbers that would be printed by the **-t**, **-a**, and **-d** flags respectively. The advantage to using the **-x** flag is that it enables an application to retrieve all of the data that describe a Problem Management subscription without having to run **pmanquery** multiple times.

Operands

None.

Description

After a Problem Management subscription definition is stored in the SDR by the **pmandef** command, you can use the **pmanquery** command to retrieve the subscription definition. The **pmanquery** command prints the details of the subscription definition in raw format which can then be parsed by other applications.

The **−n**, **−k**, **−p**, **−U** and **−H** flags control the scope of the search for subscriptions in the SDR. If none of the **−k**, **−p**, **−U** or **−H** flags is specified, the **pmanquery** command searches for Problem Management subscriptions that are owned by the end user, based on the end user's current security context. See the "Using Problem Management" chapter in the PSSP: Administration Guide for a complete description of the user's security context as in pertains to Problem Management. Refer to the Examples section for **pmanquery** flag usage information.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

The "Using the Problem Management Subsystem" chapter in *PSSP: Administration Guide*

Location

/usr/lpp/ssp/bin/pmanquery

Related Information

Commands: **pmandef**

Examples

1. To search for a subscription named my_handle that is owned by the end user, enter:

   ```
   pmanquery -n my_handle
   ```

2. To search for a subscription named my_handle that is owned by the Kerberos Version 4 principal hotdog@XYZ.COM, enter:

   ```
   pmanquery -n my_handle -k hotdog@XYZ.COM
   ```

3. To search for all subscriptions that are owned by the Kerberos Version 4 principal hotdog@XYZ.COM, enter:

   ```
   pmanquery -n all -k hotdog@XYZ.COM
   ```

4. To search for all subscriptions that are owned by the end user, enter:

   ```
   pmanquery -n all
   ```

5. To search for all subscriptions named my_handle, enter:

   ```
   pmanquery -n my_handle -k all -p all -U all -H all
   ```

6. To search for all subscriptions, enter:

   ```
   pmanquery -n all -k all -p all -U all -H all
   ```

pmanrmdloadSDR

Purpose

pmanrmdloadSDR – Reads a **pmanrmd** configuration file and loads the information into the System Data Repository (SDR).

Syntax

pmanrmdloadSDR *ConfigFileName*

Flags

None.

Operands

ConfigFileName Specifies a **pmanrmd** configuration file.

Description

The Problem Management subsystem provides 16 resource variables, named IBM.PSSP.pm.User_state1 through IBM.PSSP.pm.User_state16. These are predefined resource variables that have been set aside for system administrators to create their own resource monitors. A resource monitor that you create through Problem Management is a command that gets executed repeatedly by the **pmanrmd** daemon at a specific interval. The standard output from the command is supplied to the Event Management subsystem as the value for the resource variable. You can then use the **pmandef** command to subscribe to events for that resource variable.

The resource variable name, resource monitor command, sampling interval, and list of nodes for which the resource monitor is defined are stored in the SDR. The **pmanrmdloadSDR** command is used to store those definitions in the SDR.

You define your resource monitor to the **pmanrmd** daemon by doing the following:

- Making a copy of the **/spdata/sys1/pman/pmanrmd.conf** sample configuration file.
- Editing your copy of the configuration file. Provide the following:
  - The name of the resource variable (for example, IBM.PSSP.pm.User_state1)
  - The resource monitor command
  - A sampling interval (in seconds)
  - The nodes on which to run the resource monitor command

  When typing commands in this file, be aware that the command string of the **pmrmCommand** line will get enclosed in single-quotes and the Korn Shell rules apply to it. Keep in mind that a single quote cannot occur within single quotes. If a command contains single quotes, each must be replaced by the four characters '\'' (single quote, backslash, single quote, single quote). For example:

  ```
  Command="mount | awk '\''{print $3}'\'' | tail -1"
  ```

The marks between **awk** and **tail** are single quotes, the others are double quotes.

- Loading the configuration information into the SDR by using the **pmanrmdloadSDR** command.
- Stopping and restarting the **pmanrmd** daemon on the nodes that are affected by this change:

  Control workstation:

  ```
  stopsrc -s pmanrm.syspar_name

  startsrc -s pmanrm.syspar_name
  ```

  where *syspar_name* is the name of the system partition.

  Node:

  ```
  stopsrc -s pmanrm

  startsrc -s pmanrm
  ```

For a more complete description of Problem Management resource monitors, refer to the “Using the Problem Management Subsystem” chapter in *PSSP: Administration Guide*

Files

/spdata/sys1/pman/pmanrmd.conf
A sample **pmanrmd** configuration file.

Security

You must have write access to the SDR to run this command.

Implementation Specifics

This command is part of the IBM Parallel System Support Programs (PSSP) Licensed Program Product (LPP).

Prerequisite Information

IBM RS/6000 Cluster Technology: Event Management Programming Guide and Reference

The “Using the Problem Management Subsystem” chapter in *PSSP: Administration Guide*

Location

/usr/lpp/ssp/bin/pmanrmdloadSDR

Related Information

Commands: **pmandef**

pmv

Purpose

pmv – Specifies a parallel file move.

Syntax

| **pmv** [**−w** − | *noderange* | **'***hostlist args***'**] *mv_args*

Flags

The **pmv** command requires the first flag or parameter on the command line to be a specification of the hosts on which the command is to be executed.

| **−w** − Specifies that host names should be read from standard input. Host
| names can be in any format accepted by **rsh**.

Operands

noderange Indicates a specification via "node number." The node number corresponds to the position of a node in a frame and its slots. A node number indicates frame and slot. For example, frame 1 slot 1 would be referred to by 1. Frame 2 slot 1 would be node number 17, while frame 3 slot 2 would be 34. If a node occupies more than one slot, either node number refers to that node. Node numbers can be specified as ranges such as 1–3, which would refer to frame 1 slots 1–3, or 23–29,50,2, which would refer to frame 2 slots 7–13, frame 4 slot 2, and frame 1 slot 2. This option is only valid for an SP system.

'*hostlist args***'**
Specifies flags and arguments to be passed to the **hostlist** command. Hostlist allows several ways of listing hosts based on various criteria. Refer to the **hostlist** command.

Note: To use the working collective file specified by the WCOLL environment variable, you must specify a null string as the first argument. Refer to the **dsh** command for more information about a working collective.

mv_args Specifies arguments to the AIX **rm** or **mv** commands.

Note: The −**i** is not supported (the **dsh** command does not support standard input to remote hosts).

Description

The **pmv** command issues the AIX **mv** command on multiple hosts. The output is formatted so that duplicate output is displayed only once. The **pmv** command uses **dsh** to execute the **mv** command on multiple hosts. The output of the **ls** commands is written to standard output and formatted. The **pmv** command is identical to **pexec mv**.

Files

working collective file
See the **dsh** command.

Security

You must have access to the AIX Secure Remote Commands to run this command.

This command will automatically forward the DCE credentials if K5 is an enabled AIX authentication method and the user of the command has DCE credentials that can be forwarded. The special DCE credentials for root, called the machine or self host principal credentials, cannot be forwarded. To obtain DCE credentials that can be forwarded as a root user, a root user must issue **dce_login -f**.

Location

/usr/lpp/ssp/bin/pmv

Related Information

Commands: **dsh**, **mv**, **pexec**

Examples

To move a file from each host1, host2, and host3 to a different directory, enter:

```
pmv -w host1,host2,host3 /tmp/shnozzola /etc/shnozzola
```

ppred

Purpose

ppred – Performs a command on those hosts for which a test is satisfied.

Syntax

ppred [**–w** – | *noderange* | ' *hostlist args*'] '*ksh test*' '*true_command*' ['*false_command*']

Flags

The **ppred** command requires the first flag or parameter on the command line to be a specification of the hosts on which the command is to be executed.

–w – Specifies that host names should be read from standard input. Host names can be in any format accepted by **rsh**.

Operands

noderange Indicates a specification via "node number." The node number corresponds to the position of a node in a frame and its slots. A node number indicates frame and slot. For example, frame 1 slot 1 would be referred to by 1. Frame 2 slot 1 would be node number 17, while frame 3 slot 2 would be 34. If a node occupies more than one slot, either node number refers to that node. Node numbers can be specified as ranges such as 1–3, which would refer to frame 1 slots 1–3, or 23–29,50,2, which would refer to frame 2 slots 7–13, frame 4 slot 2, and frame 1 slot 2. This option is only valid for an SP system.

'*hostlist args***'**
Specifies flags and arguments to be passed to the **hostlist** command. Hostlist allows several ways of listing hosts based on various criteria. Refer to the **hostlist** command.

Note: To use the working collective file specified by the WCOLL environment variable, you must specify a null string as the first argument. Refer to the **dsh** command for more information about a working collective.

'*ksh test***'** **ppred** expects the second argument to be a quoted string in proper syntax to be evaluated via the **ksh test** command. This test is passed to the remote hosts and evaluated on them.

'*true_command***'**
ppred expects the third argument to be a quoted string containing a command to be executed on the hosts for which the test is true.

'*false_command***'**
ppred expects the fourth argument to be a quoted string containing a command to be executed on the hosts for which the test is false. This argument is optional.

Description

The **ppred** command performs a test on remote hosts in parallel. On each host where the test succeeds, a command is run. Optionally, a command can be specified that runs if the test is unsuccessful.

Security

You must have access to the AIX Secure Remote Commands to run this command.

This command will automatically forward the DCE credentials if K5 is an enabled AIX authentication method and the user of the command has DCE credentials that can be forwarded. The special DCE credentials for root, called the machine or self host principal credentials, cannot be forwarded. To obtain DCE credentials that can be forwarded as a root user, a root user must issue **dce_login -f** with a DCE principal other than a self host principal.

Location

/usr/lpp/ssp/bin/ppred

Related Information

Commands: **dsh**, **hostlist**, **test**

Examples

To verify that a file exists and is a regular file on the host occupying the first slot in each of 4 frames, enter:

```
ppred '-s 1-4:1' '-f /etc/passwd' 'echo \'host_name\''
```

pps

Purpose

pps – Specifies a parallel ps command.

Syntax

pps [–**w** – | *noderange* | '*hostlist args*'] *ps_args*

Flags

The **pps** command requires the first flag or parameter on the command line to be a specification of the hosts on which the command is to be executed.

–**w** – Specifies that host names should be read from standard input. Host names can be in any format accepted by **rsh**.

Operands

noderange Indicates a specification via "node number." The node number corresponds to the position of a node in a frame and its slots. A node number indicates frame and slot. For example, frame 1 slot 1 would be referred to by 1. Frame 2 slot 1 would be node number 17, while frame 3 slot 2 would be 34. If a node occupies more than one slot, either node number refers to that node. Node numbers can be specified as ranges such as 1–3, which would refer to frame 1 slots 1–3, or 23–29,50,2, which would refer to frame 2 slots 7–13, frame 4 slot 2, and frame 1 slot 2. This option is only valid for an SP system.

'*hostlist args*'
Specifies flags and arguments to be passed to the **hostlist** command. Hostlist allows several ways of listing hosts based on various criteria. Refer to the **hostlist** command.

Note: To use the working collective file specified by the WCOLL environment variable, you must specify a null string as the first argument. Refer to the **dsh** command for more information about a working collective.

ps_args Specifies arguments to the AIX **ps** command.

Description

The **pps** command uses **dsh** to execute the **ps** command on multiple hosts. The output of the **ls** commands is written to standard output and formatted so that distinct output is presented only once. The **pps** command is identical to **pexec ps**.

Files

working collective file
See the **dsh** command.

Security

You must have access to the AIX Secure Remote Commands to run this command.

This command will automatically forward the DCE credentials if K5 is an enabled AIX authentication method and the user of the command has DCE credentials that can be forwarded. The special DCE credentials for root, called the machine or self host principal credentials, cannot be forwarded. To obtain DCE credentials that can be forwarded as a root user, a root user must issue **dce_login -f**.

Location

/usr/lpp/ssp/bin/pps

Related Information

Commands: **dsh**, **pexec**, **ps**

Examples

To list processes on each host1, host2, and host3 (described previously), enter:

```
pps -w host1,host2,host3 -ef
```

preparevsd

Purpose

preparevsd – Makes a virtual shared disk available.

Syntax

preparevsd {–**a** | *vsd_name*...}

Flags

–**a** Specifies that all the virtual shared disks in the stopped state are to be prepared.

Operands

vsd_name Specifies a virtual shared disk. If the virtual shared disk is not in the stopped state, you will get an error message.

Description

The **preparevsd** command brings the specified virtual shared disks from the stopped state to the suspended state. The virtual shared disks are made available. Open and close requests are honored, while read and write requests are held until the virtual shared disks are brought to the active state. If they are in the suspended state, this command leaves them in the suspended state.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit vsd_mgmt
```

and select the Prepare a Virtual Shared Disk option.

Security

You must be in the AIX **bin** group to run this command.

Restrictions

If you have the Recoverable Virtual Shared Disk software installed and operational, do not use this command. The results may be unpredictable.

See *PSSP: Managing Shared Disks*.

Prerequisite Information

PSSP: Managing Shared Disks

Location

/usr/lpp/csd/bin/preparevsd

Related Information

Commands: **cfgvsd**, **ctlvsd**, **lsvsd**, **resumevsd**, **startvsd**, **stopvsd**, **suspendvsd**, **ucfgvsd**

Examples

To bring the virtual shared disk **vsd1vg1n1** from the stopped state to the suspended state, enter:

```
preparevsd vsd1vg1n1
```

prm

Purpose

prm – Specifies a parallel file remove.

Syntax

prm [–**w** – | *noderange* | '*hostlist args*'] *rm_args*

Flags

The **prm** command requires the first flag or parameter on the command line to be a specification of the hosts on which the command is to be executed.

–**w** – Specifies that host names should be read from standard input. Host names can be in any format accepted by **rsh**.

Operands

noderange Indicates a specification via "node number." The node number corresponds to the position of a node in a frame and its slots. A node number indicates frame and slot. For example, frame 1 slot 1 would be referred to by 1. Frame 2 slot 1 would be node number 17, while frame 3 slot 2 would be 34. If a node occupies more than one slot, either node number refers to that node. Node numbers can be specified as ranges such as 1–3, which would refer to frame 1 slots 1–3, or 23–29,50,2, which would refer to frame 2 slots 7–13, frame 4 slot 2, and frame 1 slot 2. This option is only valid for an SP system.

'*hostlist args*'
Specifies flags and arguments to be passed to the **hostlist** command. Hostlist allows several ways of listing hosts based on various criteria. Refer to the **hostlist** command.

Note: To use the working collective file specified by the WCOLL environment variable, you must specify a null string as the first argument. Refer to the **dsh** command for more information about a working collective.

rm_args Specifies arguments to the AIX **rm** command.

Description

The **prm** command issues the AIX **rm** command on multiple hosts. The output is formatted so that distinct output is displayed only once. The **prm** command uses **dsh** to execute the **rm** command on multiple hosts. The output of the **ls** commands is written to standard output and formatted. The **prm** command is identical to **pexec rm**.

Files

working collective file
See the **dsh** command.

Security

You must have access to the AIX Secure Remote Commands to run this command.

This command will automatically forward the DCE credentials if K5 is an enabled AIX authentication method and the user of the command has DCE credentials that can be forwarded. The special DCE credentials for root, called the machine or self host principal credentials, cannot be forwarded. To obtain DCE credentials that can be forwarded as a root user, a root user must issue **dce_login -f**.

Location

/usr/lpp/ssp/bin/prm

Related Information

Commands: **dsh**, **rm**, **pexec**

Examples

To remove a file from each host1, host2, and host3 (described previously), enter:

```
prm -w host1,host2,host3 /tmp/shnozzola
```

psyslclr

Purpose

psyslclr – Removes entries from **syslog** log files on a set of nodes.

Syntax

psyslclr [–**a**] [–**d** *pids*] [–**e** *endtime*] [–**f** *facilities*] [–**g** *config*]
[–**h**] [–**l** *logs*] [–**n** *nodes*] [–**p** *priority*]
[–**r** *resources*] [–**s** *startime*] [–**w** *hosts*] [–**y** *days*]

Flags

–**a** Trims logs on all nodes in the system partition.

–**d** *pids* Trims records matching the process IDs list.

–**e** *endtime* Trims records before endtime (*mmddhhmm*).

–**f** *facilities* Uses the facilities list to parse the **syslog.conf** file.

–**g** *config* Uses an alternate **syslog.conf** file.

–**h** Displays usage information.

–**l** *logs* Trims the list of log files (the **syslog.conf** file is not parsed). (This is lowercase **l**, as in **l**ist.)

–**n** *nodes* Trims records matching the nodes.

–**p** *priority* Uses priority value to parse the **syslog.conf** file.

–**r** *resources* Trims records from the resource list.

–**s** *startime* Trims records created after startime (*mmddhhmm*).

–**w** *hosts* Runs the command on a file or list of host names.

–**y** *days* Trims records more than *days* old.

Operands

None.

Description

Use this command to delete log entries in **syslogd** generated log files. Options allow for selecting the files and records that are trimmed.

The arguments to options –**d**, –**f**, –**l**, –**n**, –**r**, and –**w** can be a comma-delimited or single-quoted, blank-delimited list of values. If the –**l** flag is used, the command will only trim records from the specified list of log file names. If the –**l** flag is not passed, the command will first parse the **syslog** configuration file (the default is **/etc/syslog.conf**) to select files for trimming.

The –**f** and –**p** flags can be used to control selecting files in the configuration file. All files found in the configuration file will be trimmed if the –**f** and –**p** flags are not used.

The –**d**, –**e**, –**n**, –**r**, –**s**, and –**y** flags are used to match log entries to be deleted. A record must match a value from each of the flags that are used to be trimmed. If a flag is not passed, all records match for that field. To delete all records, use the −**y** flag with 0 as the argument. If the −**w** flag begins with a slash (/), it is interpreted as a file containing a list of nodes to execute the command on; otherwise, it can be a list as described previously. If neither the −**a** nor the −**w** flags are used, the command defaults to the local node.

Files

/etc/syslog.conf
: **syslog** daemon configuration file.

/etc/logmgt.acl
: Access Control List (ACL) file for **psyslclr** permissions.

Security

The **psyslclr** command consists of a client script and a server procedure which is executed by the Sysctl facility. Sysctl performs access authorization according to the configuration of security services on the server nodes. The server uses the Sysctl aclcheck procedure for granting access which requires the caller to have a principal entry in the log management ACL: **/etc/logmgt.acl**. The principal must log into the appropriate authentication service prior to running this command.

Location

/usr/lpp/ssp/bin/psyslclr

Related Information

Command: **psyslrpt**

Daemon: **syslogd**

Examples

1. To remove all entries older than 30 days from all **syslog** log files on all nodes in the local system partition, enter:

   ```
   psyslclr -a -y 30
   ```

2. To remove all entries between April 11th and July 23rd that were logged by ftp or snmpd on node k47n10, enter:

   ```
   psyslclr -w k47n10 -s 04110000 -e 07230000 -r ftp,snmpd
   ```

3. To remove all entries from files that may be written by user or mail facilities at a priority level of error or higher on the nodes in the **/tmp/nodelist** file, enter:

   ```
   psyslclr -w /tmp/nodelist -f mail,user -p error -y 0
   ```

psyslrpt

Purpose

psyslrpt – Generates reports of records in **syslog** log files on a set of nodes.

Syntax

psyslrpt [−**a**] [−**d** *pids*] [−**e** *endtime*] [−**f** *facilities*] [−**g** *config*]
[−**h** [−**l** *logs*] [−**n** *nodes*] [−**p** *priority*] [−**r** *resources*]
[−**s** *startime*] [−**w** *hosts*]

Flags

−**a** Generates the report on all nodes in the system partition.

−**d** *pids* Reports on records matching the process IDs list.

−**e** *endtime* Reports on records before *endtime* (*mmddhhmm*).

−**f** *facilities* Uses the facilities list to parse the **syslog.conf** file.

−**g** *config* Specifies the use of an alternate **syslog.conf** file.

−**h** Displays usage information.

−**l** *logs* Reports on the list of log files (the **syslog.conf** file is not parsed). (This is lowercase **l**, as in **l**ist.)

−**n** *nodes* Reports records matching the nodes.

−**p** *priority* Uses priority value to parse the **syslog.conf** file.

−**r** *resources* Reports records from the resource list.

−**s** *startime* Reports records created after *startime* (*mmddhhmm*).

−**w** *hosts* Runs the command on the file or list of host names.

Operands

None.

Description

Use this command to generate reports of log entries in **syslogd** generated log files. Options allow for selecting the files and records that are reported. The arguments to options −**d**, −**f**, −**l**, −**n**, −**r**, and −**w** can be a comma-delimited or single-quoted, blank-delimited list of values. If the −**l** flag is used, the command will report records from the specified list of log file names. If the −**l** flag is not passed, the command will first parse the **syslog** configuration file (the default is **/etc/syslog.conf**) to select files for reporting.

The −**f** and −**p** options can be used to control the selecting of files in the configuration file. All files found in the configuration file are reported on if the −**f** and −**p** flags are not used.

The −**d**, −**e**, −**n**, −**r**, and −**s** options are used to match log entries to be reported. A record must match a value from each of these flags that are used to be reported. If a flag is not passed, all records match for that field. If the −**w** argument begins with slash (/), it is interpreted as a file containing a list of nodes to execute the command on; otherwise, it can be a list as described previously. If neither the −**a** nor −**w** flags are used, the command defaults to the local node.

Files

/etc/syslog.conf
syslog daemon configuration file.

Security

The **psyslrpt** command consists of a client script and a server procedure which is executed by the Sysctl facility. Sysctl callbacks perform access authorization according to the configuration of security services on the server nodes. The server procedure uses the Sysctl AUTH callback for granting access which requires the caller to have been authenticated in accordance with the security policy of the target node or nodes.

Location

/usr/lpp/ssp/bin/psyslrpt

Related Information

Command: **psyslclr**

Daemon: **syslogd**

The *PSSP: Administration Guide*

Examples

1. To report all entries from all **syslog** log files on all nodes in the local system partition starting on March 3rd, enter:

```
psyslrpt -a -s 03030000
```

2. To report all entries between April 11th and July 23rd that were logged by ftp or snmpd on node k47n10, enter:

```
psyslrpt -w k47n10 -s 04110000 -e 07230000 -r ftp,snmp
```

3. To report entries from the specific log file **/var/adm/SPlogs/SPdaemon.log** with process IDs 10479 or 1157 on nodes k47n12 and k47n15, enter:

```
psyslrpt -w k47n12,k47n15 -d'10479 1157' -l /var/adm/SPlogs/SPdaemon.log
```

rcmdtgt

Purpose

rcmdtgt – Obtains a Kerberos Version 4 authentication ticket for the local realm, with a maximum allowed lifetime, using the service key for the instance of the **rcmd** principal on the local host.

Syntax

rcmdtgt

Flags

None.

Operands

None.

Description

Use this command to obtain Kerberos Version 4 authorization with a maximum allowed lifetime, using the service key for **rcmd.***localhost* found in the service key file at **/etc/krb-srvtab**. When using SP authentication services, these tickets have an unlimited lifetime. When using AFS authentication services, a maximum of 30 days is enforced.

This command is intended primarily for use in shell scripts and other batch-type facilities.

The KRBTKFILE environment variable must be used to specify the ticket cache file used by **rcmdtgt** to store authentication tickets.

If Kerberos Version 4 is not an active authentication method for AIX remote commands, and Compatibility is not an active authentication method for SP trusted services, this command performs no function but returns successfully.

Because the ticket obtained using this command may not expire, the user should be careful to delete the temporary ticket file.

When using **/usr/lpp/ssp/rcmd/bin/rcmdtgt**, remember to check that the authentication method is in fact Kerberos Version 4 before using **k4destroy** or **/usr/lpp/ssp/kerberos/bin/kdestroy** to destroy credentials. While Kerberos Version 4 may be configured, the authentication method may be superseded by DCE and you could be destroying credentials obtained by the system administrator through a Kerberos Version 4 login.

Environment Variables

KRBTKFILE
The pathname of the ticket cache file to use.

Files

/etc/krb.conf Contains the name of the local realm.

/etc/krb-srvtab Specifies the service key file.

Security

You must have root privilege to run this command.

Location

/usr/lpp/ssp/rcmd/bin/rcmdtgt

Related Information

Commands: **k4destroy**, **k4init**

File: **krb.conf**

Refer to the "RS/6000 SP Files and Other Technical Information" section of *PSSP: Command and Technical Reference* for additional **Kerberos** information.

Examples

The following example, excerpted from the sample **script.cust** file, shows how **rcmdtgt** can be used in a shell script to perform the authentication required to use the **rcp** command:

```
# set the host name from which you will copy the file.

SERVER='cat /etc/ssp/server_host_name | cut -d" " -f1'



# Define a temporary ticket cache file, then get a ticket

export KRBTKFILE=/tmp/tkt.$$

/usr/lpp/ssp/rcmd/bin/rcmdtgt

#

# Perform kerberos-authenticated rcp

# rcp is linked to AIX rcp

rcp $SERVER:/etc/resolv.conf /etc/resolv.conf



# Remove the ticket cache file

/bin/k4destroy

unset KRBTKFILE
```

removehsd

Purpose

removehsd – Removes one or more hashed shared disks, the virtual shared disks associated with them, and the System Data Repository (SDR) information for virtual shared disks on the associated nodes.

Syntax

removehsd {−**v** *hsd_names* | −**a**} [−**f**]

Flags

−v Specifies the hashed shared disk name or names that are to be removed by this command.

−a Specifies that the command should remove all hashed shared disks in the system or system partition.

−f Forces the system to unconfigure the hashed shared disks and its underlying virtual shared disks and remove them. If −**f** is not specified and any of the virtual shared disks that constitute the hashed shared disks to be removed are configured or the hashed shared disk itself is configured, the command is unsuccessful.

Operands

None.

Description

Use this command to remove the logical volumes associated with virtual shared disks in the set of hashed shared disks. The order in which the virtual shared disks that make up the hashed shared disks and the hashed shared disks themselves are removed is the reverse of the order in which they were created.

If the virtual shared disk or hashed shared disk is configured on any of the nodes on the system partition, this command is unsuccessful, unless the −**f** flag is specified.

Security

You must have access to the virtual shared disk subsystem via the sysctl service to run this command.

Prerequisite Information

PSSP: Managing Shared Disks

Location

/usr/lpp/csd/bin/removehsd

Related Information

Commands: **createhsd**, **removevsd**

Examples

To unconfigure and remove the virtual shared disks associated with the hashed shared disks DATA and remove the hashed shared disk as well, type:

```
removehsd -d DATA -f
```

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit delete_vsd
```

and select the Remove a Hashed Shared Disk option.

removevsd

Purpose

removevsd – Removes a set of virtual shared disks that are not part of any hashed shared disk.

Syntax

removevsd {–**v** *vsd_names* | –**a**} [–**f**]

Flags

–**v** Specifies the virtual shared disk name or names that are to be removed by this command.

–**a** Specifies that the command should remove all virtual shared disks in the system or system partition.

–**f** Forces the system to unconfigure the virtual shared disks and remove them. If –**f** is not specified and any of the virtual shared disks that are to be removed are configured, the command is unsuccessful.

Operands

None.

Description

Use this command to remove the logical volumes associated with the virtual shared disks and update the backup nodes' Object Data Managers (ODMs), if any exist. The virtual shared disk information will be deleted from the System Data Repository (SDR). The removal of the virtual shared disks is done in the reverse of the order in which they were created. Volume groups are not removed with this command.

If the virtual shared disk is configured on any of the nodes on the system partition, this command is unsuccessful, unless the –**f** flag is specified.

Note: This command is unsuccessful if one of the virtual shared disks named in *vsd_names* belongs to a hashed shared disk. To remove virtual shared disks that belong to a hashed shared disk, use **removehsd**.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit delete_vsd
```

and select the Remove a Virtual Shared Disk option.

Security

You must have access to the virtual shared disk subsystem via the sysctl service to run this command.

Prerequisite Information

PSSP: Managing Shared Disks

Related Information

Commands: **createvsd**, **removehsd**

Location

/usr/lpp/csd/bin/removevsd

Examples

To unconfigure and remove all defined virtual shared disks in a system or system partition, enter:

```
removevsd -a -f
```

resource_center

Purpose

resource_center – Invokes the RS/6000 SP Resource Center.

Syntax

resource_center [−**c**] [**Netscape flags ...**]

Flags

−**c** Forces the SP Resource Center to prompt the user for the Netscape location.

Netscape flags
Refer to Netscape documentation or run **netscape** −**h** to list available flags.

Operands

None.

Description

The RS/6000 SP Resource Center provides one single interface for all softcopy SP documentation and information resources. It consists of HTML files, Java and JavaScript, and runs in Netscape Navigator. The SP Resource Center provides access to the following RS/6000 SP information:

- Publications, READMEs, Redbooks, White Papers
- Up-to-date Service Information
- SP Product Information (Software & Hardware)
- Many other online resources useful to the RS/6000 SP user and administrator.

Upon invoking the **resource_center** command for the first time, a dialog box will ask you for the location of the Netscape executable that is installed on your system. Enter the full pathname to the Netscape program (for example, **/usr/local/bin/netscape**). This path information is stored in your **$HOME/.resctr** file, and you will only be prompted for it once.

The **resource_center** command will bring up Netscape Navigator with the top level RS/6000 SP Resource Center page loaded. There are three frames on the SP Resource Center interface. The frame on the top is the Title frame. The frame on the left is the Navigation frame. The large frame on the right is the Display frame.

The Title frame lets you access the IBM Home Page on the Internet (click on the IBM logo), go back to the top of the SP Resource Center (the "Home" link), search the contents of the SP Resource Center (the "Search" link), display an index of the SP Resource Center's contents (the "Index" link), and obtain help about the use of the SP Resource Center (the "Help" link).

The Navigation frame lets you select content to view. This frame contains categories that may be expanded and collapsed to display their sub-categories. Click on a category with a right-pointing arrow to expand the category, and click on a category with a down-pointing arrow to collapse the category. When one category is expanded, all other categories are collapsed. When a category is expanded, any of the sub-categories may be selected, and the contents are displayed in the Display frame.

Some sub-categories on the Navigation frame include a small "world" icon that indicates that the link will take you to the Internet. If you do not have an internet connection, these links will not work. When you select a link to the internet, a new Netscape window appears. This ensures that the SP Resource Center Title and Navigation frames do not get in the way of the internet web page.

The Display frame is used to display all local information. When a non-internet link is selected from the Navigation frame, the resulting information is displayed in the Display frame.

To quit the SP Resource Center, exit Netscape Navigator.

Environment Variables

The environment variable **NETSCAPE** is used (if set) to specify the pathname to the Netscape Navigator web browser that will be used to display the RS/6000 SP Resource Center.

Files

$HOME/.resctr Stores the pathname to Netscape Navigator for each user.

Restrictions

If your machine does not have a connection to the internet, some of the SP Resource Center's hyperlinks will not function.

Web Pages on the internet that the SP Resource Center points to may not be available due to the dynamic nature of the web.

Prerequisite Information

Netscape Navigator version 4 or later is required to run this command. The first time each user invokes the **resource_center** command, the pathname to the Netscape program is stored in **$HOME/.resctr**.

If online publications are installed on the system, the SP Resource Center will use the local copies, otherwise it will look on the web for the online publications.

Location

/usr/lpp/ssp/bin/resource_center

Examples

To invoke the RS/6000 SP Resource Center, enter:

```
resource_center
```

This example assumes the directory **/usr/lpp/ssp/bin** is in your path.

resumevsd

Purpose

resumevsd – Activates an available virtual shared disk.

Syntax

resumevsd[–**p** | –**b** | –**l** *server_list*] {–**a** | *vsd_name* ...}

Flags

–**p** Specifies that the primary server node defined for the global volume group is to be the active server. The –**p** flag is not valid for CVSD.

–**b** Specifies that the secondary server node defined for the global volume group is to be the active server. The –**b** flag is not valid for CVSD.

Note: This flag is used only by the Recoverable Virtual Shared Disk subsystem.

–**a** Specifies that all the virtual shared disks that have been defined are to be resumed.

–**l** Passes the **server_list** to the driver.

Operands

vsd_name Specifies a virtual shared disk.

Description

The **resumevsd** command brings the specified virtual shared disks from the suspended state to the active state. The virtual shared disks remains available. Read and write requests which had been held while the virtual shared disk was in the suspended state are resumed.

You can use the System Management Interface Tool (SMIT) to run this command. To use SMIT, enter:

```
smit vsd_mgmt
```

and select the Resume a Virtual Shared Disk option.

Security

You must be in the AIX **bin** group to run this command.

Restrictions

1. If you have the Recoverable Virtual Shared Disk software installed and operational, do not use this command. The results may be unpredictable.

   See *PSSP: Managing Shared Disks*

2. The –**b** flag is used only by the Recoverable Virtual Shared Disk subsystem.

Prerequisite Information

PSSP: Managing Shared Disks

Location

/usr/lpp/csd/bin/resumevsd

Related Information

Commands: **cfgvsd**, **ctlvsd**, **lsvsd**, **preparevsd**, **startvsd**, **stopvsd**, **suspendvsd**, **ucfgvsd**

Examples

To bring the virtual shared disk **vsd1vg1n1** from the suspended state to the active state, enter:

```
resumevsd vsd1vg1n1
```

rm_spsec

Purpose

rm_spsec – Removes configuration from DCE service principals and keyfiles.

Syntax

rm_spsec [−**h**] [−**v**] [−**p**] −**t** {*local* | *admin*} [*dce_hostname* | *partition_name*]

Flags

−h Prints command syntax to standard output.

−v Prints progress messages to standard output.

−t When *local* is specified, the command removes local keyfiles and keytab objects for the host. When *admin* is specified, all SP Trusted Services principals, accounts, and rpcentries for the specified DCE hostname are removed from the DCE registry.

−p Specifies the partition name. The *partition_name* operand is required.

Operands

dce_hostname **|** *partition_name*
This operand is an option when using −**t** *admin*. The −**p** flag specifies a partition name; otherwise this is a DCE hostname.

Description

The **rm_spsec** command reads from two files, a default file (**/usr/lpp/ssp/config/spsec_defaults**) and an override file (**/spdata/sys1/spsec/spsec_overrides**). The two types of configuration removal are *admin* and *local*.

The *local* version of this command must be executed on the host which is to have its definition and use of SP Security Services removed from DCE registry, and must be run prior to running the *admin* version of the command.

Issuing **rm_spsec** −**t** *local* will delete keytab objects for the current host. Removing keytab objects also removes the keyfiles from the **/spdata/sys1/keyfiles** directory. The **sysctl** ACL files will be deleted from the **/var/sysctl** directory. If this command is run on the control workstation and *local* is used, the **hardmon** ACL files will be deleted from the **/spdata/sys1/spmon/hmdceacls** directory.

The *admin* version of this command requires cell administration authority and access to the DCE registry. When −**t** *admin* is specified, all SP Trusted Services principals (including their entries in any groups and organizations), accounts, rpcentries (including their directories) for the specified DCE hostname are removed from the DCE registry.

The −**p** flag specifies a partition name for either *local* or *admin*. For *local*, the command must be run on the control workstation (because that is where the partition sensitive keyfiles reside).

For syntax errors within either of the input files, an error message is issued and logged and processing halts. All errors are logged to the **/var/SPlogs/auth_install/log** and printed to stdout.

Deletion of data from the CDS is a destructive and non-retrievable process. Prior to running this command, IBM suggests backing up any relevant DCE databases.

Files

input: **/usr/lpp/ssp/config/spsec_defaults**

/spdata/sys1/spsec/spsec_overrides

output: Log file created: **/var/adm/SPlogs/auth_install/log**

CDS registry and Security Server database updated

Exit Values

0 Indicates successful completion of the command.

1 Indicates that errors occurred during the execution of this program. Review any reported errors either on the console or in the Log file.

Information pertaining to the specific workstation's service principals and account information may remain in the Security registry and the associated keyfiles may remain on the system.

Security

You must have DCE cell administrator authority for removal of *admin* configuration, and root authority for removal of *local* configuration.

Location

/usr/lpp/ssp/bin/rm_spsec

Related Information

File: **/usr/lpp/ssp/config/spsec_defaults**

File: **/spdata/sys1/spsec/spsec_overrides**

DCE Administration Publications for AIX

Commands: **config_spsec**, **create_keyfiles**

Examples

1. To remove keytab objects for a specified host or to unconfigure SP Trusted Services for a specified host, enter:

   ```
   rm_spsec -v -t local
   ```

   Then log into the DCE cell with an ID that has cell administration authority, and enter:

   ```
   rm_spsec -v -t admin melmac.pok.ibm.com
   ```

2. To remove keytab objects and keyfiles for partition named "alf99" as root user, enter:

```
rm_spsec -v -t local -p alf99
```

rmkp

Purpose

rmkp – Removes Kerberos Version 4 principals.

Syntax

rmkp −**h**

rmkp [−**n**] [−**v**] {*name*[*.instance*]|*name.*|*.instance*} ...

Flags

−**h** Displays usage information.

−**n** Suppresses prompting for confirmation.

−**v** Specifies verbose mode (displays informational messages).

Operands

{*name***[***.instance***]|***name.***|***.instance***} ...**
Identifies specific principals to remove. When the command is invoked interactively (without the −**n** flag and not through Sysctl), you can use special notation to select all principals with a particular name or instance that you want to remove. Specify *name.* to remove all principals with a specific name or *.instance* to remove all principals with a specific instance.

Note: The *name* must be followed by a period and the *instance* must be preceded by a period.

Description

Use this command to remove principals from the local Kerberos Version 4 database. You will be prompted to confirm each deletion prior to its execution. This command will not remove any of the four principals that were predefined by Kerberos Version 4 when the database was created. Deleted entries are saved in the **/var/kerberos/database/rmkp.save.**<**PID**> file, in the readable ASCII format produced by the **kdb_util dump** command. The **rmkp** command should normally be used only on the primary server. If there are secondary authentication servers, the **push-kprop** command is invoked to propagate the change to the other servers. The command can be used to update a secondary server's database, but the changes may be negated by a subsequent update from the primary.

Files

/var/kerberos/database/admin_acl.add
Access control list for **kadmin**, **mkkp**, and **rmkp**.

/var/kerberos/database/principal.*
Kerberos database files.

/var/kerberos/database/rmkp.save.<**pid**>
File containing removed Kerberos database entries.

Standard Output

When the −**v** option is omitted, only the prompt for confirmation is written to standard output. When the −**v** flag is specified, the disposition of each selected principal is indicated by a message, and the name of the file containing the removed entries is printed. The −**v** flag has no effect on error messages written to standard error.

Exit Values

0 Indicates the successful completion of the command. At least one principal was found that matched the specified names. Whether or not any were removed depends on the responses you entered when prompted. If you entered a principal that does not exist, or if you entered an operand of the form *name.* or *.instance* in noninteractive mode, a message is written to standard error and processing continues with any remaining principals.

1 Indicates that an error occurred and no principal was removed. One of the following conditions was detected:

- The command was incorrectly specified with no operand or a flag that is not valid.
- No principal was found matching the names specified.
- The host on which the command was issued is not an authentication server.
- The database was changed by another process while this command was executing.
- The **kdb_util** command was unsuccessful.

Security

You must have root privilege and be logged on to a Kerberos Version 4 server host. It can be invoked indirectly as a Sysctl procedure by a Kerberos database administrator who has a valid ticket and is listed in the **admin_acl.add** file.

Restrictions

When you execute the **rmkp** command through the Sysctl procedure of the same name, the −**n** flag is added to your command invocation. This is required because Sysctl does not provide an interactive environment that supports prompting for confirmation. Suppressing confirmation increases the risk of unintentionally removing the wrong principal. In this mode, each principal to be removed must be named explicitly; selection of multiple principals by name or instance alone is not allowed. Since nonroot Kerberos administrators can execute this command only through Sysctl, you must be root on the server to use the special notation for selecting multiple principals.

Location

/usr/kerberos/etc/rmkp

Related Information

Commands: **chkp**, **kadmin**, **kdb_util**, **lskp**, **mkkp**, **sysctl**

Examples

1. To remove Kerberos principal tempuser, enter:

   ```
   rmkp tempuser
   ```

   You should receive a prompt similar to the following:

   ```
   Confirm removal of principal tempuser? (y or n): y
   ```

2. To remove (be given the option to remove) all instances of joe, frank, and the rcmd service principal with instance node 25tr, enter:

   ```
   rmkp -v joe. frank rcmd.node25tr
   ```

   You should receive prompts similar to the following:

   ```
   Confirm removal of principal joe? (y or n): n

   joe was not removed

   Confirm removal of principal joe.admin? (y or n): y

   joe.admin was removed

   Confirm removal of principal frank? (y or n): y

   frank was removed

   Confirm removal of principal rcmd.node25tr? (y or n): y

   rcmd.node25tr was removed

   Removed entries were saved in /var/kerberos/database/rmkp.save.7942
   ```

rvsdrestrict

Purpose

rvsdrestrict – Displays and sets which level of the IBM Recoverable Virtual Shared Disk software is to run when you have a system partition with mixed levels of the PSSP or IBM Recoverable Virtual Shared Disk software.

Syntax

rvsdrestrict {**-l** | **-s** {RVSD1.2 | RVSD2.1 | RVSD3.1 | RESET }}

Flags

–l Lists the current rvsd subsystem run level as recorded in the SDR.

–s Sets the rvsd subsystem run level to RVSD 1.2, RVSD 2.1, or RVSD 3.1 in the SDR, or resets the rvsd subsystem run level.

Operands

None.

Description

The **rvsdrestrict** command is used to restrict the level that the IBM Recoverable Virtual Shared Disk software will run at. This command must be used when in a system partition with mixed levels of PSSP and or mixed levels of the IBM Recoverable Virtual Shared Disk software. If a node has a lower level of the IBM Recoverable Virtual Shared Disk software installed than what is set with this command, then the rvsd subsystem will not start on that node.

This command does not dynamically change rvsd subsystem run levels across the SP. An rvsd subsystem instance will only react to this information after being restarted. Thus, if your cluster runs at a given level, and you want to override this level you must stop rvsd subsystem on all nodes, override the level, and restart.

Standard Output

Current rvsd subsystem run level as recorded in the SDR.

Security

You must be in the AIX **bin** group and have write access to the SDR to run this command.

Location

/usr/lpp/csd/bin/rvsdrestrict

Examples

1. If you were to have nodes with RVSD1.2, RVSD2.1, and RVSD3.1 all in the same system partition, and you wanted them to all run and coexist, you would issue:

   ```
   rvsdrestrict -s RVSD1.2
   ```

This will force all the rvsd subsystems to run at the functionality level of RVSD1.2.

2. If you were to have nodes with RVSD1.2, RVSD2.1, and RVSD3.1 all in the same system partition, but you only wanted the rvsd subsystem to start on nodes that were capable of running RVSD3.1, then you would issue:

```
rvsdrestrict -s RVSD3.1
```

Part 2. Appendixes

Appendix A. Perspectives Colors and Fonts

Perspectives Colors with Red, Green, and Blue (RGB) Triplets

The following list contains valid color names that can be supplied as optional arguments to the –**backgroundColor** and –**foregroundColor** flags. Colors may vary depending on the type of display you are using.

Color	Red	Green	Blue
aquamarine	127	255	212
azure	240	255	255
beige	245	245	220
bisque	255	228	196
black	0	0	0
blue	0	0	255
brown	165	42	42
burlywood	222	184	135
chartreuse	127	255	0
chocolate	210	105	30
coral	255	127	80
cornsilk	255	248	220
cyan	0	255	255
firebrick	178	34	34
gold	255	215	0
goldenrod	218	165	32
gray	190	190	190
green	0	255	0
honeydew	240	255	240
ivory	255	255	240
khaki	240	230	140
lavender	230	230	250
linen	250	240	230
magenta	255	0	255
maroon	176	48	96
moccasin	255	228	181
oldlace	253	245	230
orange	255	165	0
orchid	218	112	214
peru	205	133	63
pink	255	192	203
plum	221	160	221
purple	160	32	240
red	255	0	0
salmon	250	128	114
seashell	255	245	238
sienna	160	82	45
snow	255	250	250
tan	210	180	140
thistle	216	191	216
tomato	255	99	71
turquoise	64	224	208
violet	238	130	238
wheat	245	222	179
white	255	255	255
yellow	255	255	0

Perspectives Fonts

Note: Fonts will vary depending on the type of Xmachine or Xstation you are using.

The following list contains font names that can be supplied as optional arguments to the −**fontFamily** flag:

application
block
charter
clean
courier
ergonomic
fixed
helvetica
lucida
lucida bright
lucida typewriter
new century schoolbook
roman
sans serif
serif
special
terminal
times
times new roman
type
typewriter
utopia

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department LJEB/P905
| 2455 South Road
Poughkeepsie, NY 12601-5400
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States or other countries or both:

AIX
AIX/6000
DATABASE 2
DB2
ES/9000
ESCON
HACMP/6000
IBM
IBMLink
LoadLeveler
Micro Channel
Netfinity
Netfinity Manager
POWERparallel
POWERserver
RS/6000
RS/6000 Scalable POWERparallel Systems
Scalable POWERparallel Systems
SP
System/370
System/390
TURBOWAYS

| Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

| Microsoft, Windows, Windows NT, BackOffice, MS-DOS, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Tivoli Enterprise Console is a trademark of Tivoli Systems Inc. in the United States, other countries, or both.

| UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

Other company, product, and service names may be the trademarks or service marks of others.

Publicly Available Software

PSSP includes software that is publicly available:

expect Programmed dialogue with interactive programs

Perl Practical Extraction and Report Language

SUP Software Update Protocol

Tcl Tool Command Language

TclX Tool Command Language Extended

Tk Tcl-based Tool Kit for X-windows

This book discusses the use of these products only as they apply specifically to the RS/6000 SP system. The distribution for these products includes the source code and associated documentation. **/usr/lpp/ssp/public** contains the compressed **tar** files of the publicly available software. (IBM has made minor modifications to the versions of Tcl and Tk used in the SP system to improve their security characteristics. Therefore, the IBM-supplied versions do not match exactly the versions you may build from the compressed **tar** files.) All copyright notices in the documentation must be respected. You can find version and distribution information for each of these products that are part of your selected install options in the **/usr/lpp/ssp/READMES/ssp.public.README** file.

Software Update Protocol

| IBM has provided modifications to this software. The resulting software is provided to you on an “AS IS” basis and WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Glossary of Terms and Abbreviations

A

ACL. Access Control List. A list that defines who has permission to access certain services; that is, for whom a server may perform certain tasks. This is usually a list of principals with the type of access assigned to each.

adapter. An adapter is a mechanism for attaching parts. For example, an adapter could be a part that electrically or physically connects a device to a computer or to another device. In the SP system, network connectivity is supplied by various adapters, some optional, that can provide connection to I/O devices, networks of workstations, and mainframe networks. Ethernet, FDDI, token-ring, HiPPI, SCSI, FCS, and ATM are examples of adapters that can be used as part of an SP system.

address. A character or group of characters that identifies a register, a device, a particular part of storage, or some other data source or destination.

AFS. A distributed file system that provides authentication services as part of its file system creation.

AIX. Abbreviation for Advanced Interactive Executive, IBM's licensed version of the UNIX operating system. AIX is particularly suited to support technical computing applications, including high function graphics and floating point computations.

Amd. Berkeley Software Distribution automount daemon.

API. Application Programming Interface. A set of programming functions and routines that provide access between the Application layer of the OSI seven-layer model and applications that want to use the network. It is a software interface.

application. The use to which a data processing system is put; for example, a payroll application, an airline reservation application.

application data. The data that is produced using an application program.

ARP. Address Resolution Protocol.

ATM. Asynchronous Transfer Mode. (See *TURBOWAYS 100 ATM Adapter*.)

authentication. The process of validating the identity of either a user of a service or the service itself. The process of a principal proving the authenticity of its identity.

authorization. The process of obtaining permission to access resources or perform tasks. In SP security services, authorization is based on the principal identifier. The granting of access rights to a principal.

authorization file. A type of ACL (access control list) used by the IBM AIX remote commands and the IBM PSSP Sysctl and Hardmon components.

B

batch processing. * (1) The processing of data or the accomplishment of jobs accumulated in advance in such a manner that each accumulation thus formed is processed or accomplished in the same run. * (2) The processing of data accumulating over a period of time. * (3) Loosely, the execution of computer programs serially. (4) Computer programs executed in the background.

BMCA. Block Multiplexer Channel Adapter. The block multiplexer channel connection allows the RS/6000 to communicate directly with a host System/370 or System/390; the host operating system views the system unit as a control unit.

BOS. The AIX Base Operating System.

C

call home function. The ability of a system to call the IBM support center and open a PMR to have a repair scheduled.

CDE. Common Desktop Environment. A graphical user interface for UNIX.

charge feature. An optional feature for either software or hardware for which there is a charge.

CLI. Command Line Interface.

client. * (1) A function that requests services from a server and makes them available to the user. * (2) A term used in an environment to identify a machine that uses the resources of the network.

Client Input/Output Sockets (CLIO/S). A software package that enables high-speed data and tape access between SP systems, AIX systems, and ES/9000 mainframes.

CLIO/S. Client Input/Output Sockets.

CMI. Centralized Management Interface provides a series of SMIT menus and dialogues used for defining and querying the SP system configuration.

Concurrent Virtual Shared Disk. A virtual shared disk that can be concurrently accessed by more than one server.

connectionless. A communication process that takes place without first establishing a connection.

connectionless network. A network in which the sending logical node must have the address of the receiving logical node before information interchange can begin. The packet is routed through nodes in the network based on the destination address in the packet. The originating source does not receive an acknowledgment that the packet was received at the destination.

control workstation. A single point of control allowing the administrator or operator to monitor and manage the SP system using the IBM AIX Parallel System Support Programs.

credentials. A protocol message, or part thereof, containing a ticket and an authenticator supplied by a client and used by a server to verify the client's identity.

css. Communication subsystem.

D

daemon. A process, not associated with a particular user, that performs system-wide functions such as administration and control of networks, execution of time-dependent activities, line printer spooling and so forth.

DASD. Direct Access Storage Device. Storage for input/output data.

DCE. Distributed Computing Environment.

DFS. distributed file system. A subset of the IBM Distributed Computing Environment.

DNS. Domain Name Service. A hierarchical name service which maps high level machine names to IP addresses.

E

Error Notification Object. An object in the SDR that is matched with an error log entry. When an error log entry occurs that matches the Notification Object, a user-specified action is taken.

ESCON. Enterprise Systems Connection. The ESCON channel connection allows the RS/6000 to communicate directly with a host System/390; the host operating system views the system unit as a control unit.

Ethernet. (1) Ethernet is the standard hardware for TCP/IP local area networks in the UNIX marketplace. It is a 10-megabit per second baseband type LAN that allows multiple stations to access the transmission medium at will without prior coordination, avoids contention by using carrier sense and deference, and resolves contention by collision detection (CSMA/CD). (2) A passive coaxial cable whose interconnections contain devices or components, or both, that are all active. It uses CSMA/CD technology to provide a best-effort delivery system.

Ethernet network. A baseband LAN with a bus topology in which messages are broadcast on a coaxial cabling using the carrier sense multiple access/collision detection (CSMA/CD) transmission method.

event. In Event Management, the notification that an expression evaluated to true. This evaluation occurs each time an instance of a resource variable is observed.

expect. Programmed dialogue with interactive programs.

expression. In Event Management, the relational expression between a resource variable and other elements (such as constants or the previous value of an instance of the variable) that, when true, generates an event. An example of an expression is `X < 10` where X represents the resource variable `IBM.PSSP.aixos.PagSp.%totalfree` (the percentage of total free paging space). When the expression is true, that is, when the total free paging space is observed to be less than 10%, the Event Management subsystem generates an event to notify the appropriate application.

F

failover. Also called fallover, the sequence of events when a primary or server machine fails and a secondary or backup machine assumes the primary workload. This is a disruptive failure with a short recovery time.

fall back. Also called fallback, the sequence of events when a primary or server machine takes back control of its workload from a secondary or backup machine.

FDDI. Fiber Distributed Data Interface.

| **FFDC**. First Failure Data Capture.

Fiber Distributed Data Interface (FDDI). An American National Standards Institute (ANSI) standard for 100-megabit-per-second LAN using optical fiber cables. An FDDI local area network (LAN) can be up to 100 km (62 miles) and can include up to 500 system units. There can be up to 2 km (1.24 miles) between system units and concentrators.

file. * A set of related records treated as a unit, for example, in stock control, a file could consist of a set of invoices.

file name. A CMS file identifier in the form of 'filename filetype filemode' (like: TEXT DATA A).

file server. A centrally located computer that acts as a storehouse of data and applications for numerous users of a local area network.

File Transfer Protocol (FTP). The Internet protocol (and program) used to transfer files between hosts. It is an application layer protocol in TCP/IP that uses TELNET and TCP protocols to transfer bulk-data files between machines or hosts.

| **First Failure Data Capture (FFDC)**. A set of utilities used for recording persistent records of failures and significant software incidents. It provides a means of associating failures to one another, thus allowing software to link effects of a failure to their causes and thereby facilitating discovery of the root cause of a failure.

foreign host. Any host on the network other than the local host.

FTP. File transfer protocol.

G

gateway. An intelligent electronic device interconnecting dissimilar networks and providing protocol conversion for network compatibility. A gateway provides transparent access to dissimilar networks for nodes on either network. It operates at the session presentation and application layers.

H

HACMP. High Availability Cluster Multi-Processing for AIX.

HACWS. High Availability Control Workstation function, based on HACMP, provides for a backup control workstation for the SP system.

HAL. Hardware Abstraction Layer, a communication device interface that provides communication channels for processes.

Hashed Shared Disk (HSD). The data striping device for the IBM Virtual Shared Disk. The device driver lets application programs stripe data across physical disks in multiple IBM Virtual Shared Disks, thus reducing I/O bottlenecks.

help key. In the SP graphical interface, the key that gives you access to the SP graphical interface help facility.

High Availability Cluster Multi-Processing. An IBM facility to cluster nodes or components to provide high availability by eliminating single points of failure.

HiPPI. High Performance Parallel Interface. RS/6000 units can attach to a HiPPI network as defined by the ANSI specifications. The HiPPI channel supports burst rates of 100 Mbps over dual simplex cables; connections can be up to 25 km in length as defined by the standard and can be extended using third-party HiPPI switches and fiber optic extenders.

home directory. The directory associated with an individual user.

host. A computer connected to a network, and providing an access method to that network. A host provides end-user services.

I

instance vector. Obsolete term for resource identifier.

Intermediate Switch Board. Switches mounted in the switch expansion frame.

Internet. A specific inter-network consisting of large national backbone networks such as APARANET, MILNET, and NSFnet, and a myriad of regional and campus networks all over the world. The network uses the TCP/IP protocol suite.

Internet Protocol (IP). (1) A protocol that routes data through a network or interconnected networks. IP acts as an interface between the higher logical layers and the physical network. This protocol, however, does not provide error recovery, flow control, or guarantee the reliability of the physical network. IP is a connectionless protocol. (2) A protocol used to route data from its source to it destination in an Internet environment.

IP address. A 32-bit address assigned to devices or hosts in an IP internet that maps to a physical address. The IP address is composed of a network and host portion.

ISB. Intermediate Switch Board.

K

Kerberos. A service for authenticating users in a network environment.

kernel. The core portion of the UNIX operating system which controls the resources of the CPU and allocates them to the users. The kernel is memory-resident, is said to run in "kernel mode" and is protected from user tampering by the hardware.

| **Kernel Low-Level Application Programming
| Interface (KLAPI)**. KLAPI provides transport service
| for communication using the SP Switch.

L

LAN. (1) Acronym for Local Area Network, a data network located on the user's premises in which serial transmission is used for direct data communication among data stations. (2) Physical network technology that transfers data a high speed over short distances. (3) A network in which a set of devices is connected to another for communication and that can be connected to a larger network.

local host. The computer to which a user's terminal is directly connected.

log database. A persistent storage location for the logged information.

log event. The recording of an event.

log event type. A particular kind of log event that has a hierarchy associated with it.

logging. The writing of information to persistent storage for subsequent analysis by humans or programs.

M

mask. To use a pattern of characters to control retention or elimination of portions of another pattern of characters.

menu. A display of a list of available functions for selection by the user.

Motif. The graphical user interface for OSF, incorporating the X Window System. Also called OSF/Motif.

MTBF. Mean time between failure. This is a measure of reliability.

MTTR. Mean time to repair. This is a measure of serviceability.

N

naive application. An application with no knowledge of a server that fails over to another server. Client to server retry methods are used to reconnect.

network. An interconnected group of nodes, lines, and terminals. A network provides the ability to transmit data to and receive data from other systems and users.

NFS. Network File System. NFS allows different systems (UNIX or non-UNIX), different architectures, or vendors connected to the same network, to access remote files in a LAN environment as though they were local files.

NIM. Network Installation Management is provided with AIX to install AIX on the nodes.

NIM client. An AIX system installed and managed by a NIM master. NIM supports three types of clients:

- Standalone
- Diskless
- Dataless

NIM master. An AIX system that can install one or more NIM clients. An AIX system must be defined as a NIM master before defining any NIM clients on that system. A NIM master managers the configuration database containing the information for the NIM clients.

NIM object. A representation of information about the NIM environment. NIM stores this information as objects in the NIM database. The types of objects are:

- Network
- Machine
- Resource

NIS. Network Information System.

node. In a network, the point where one or more functional units interconnect transmission lines. A computer location defined in a network. The SP system can house several different types of nodes for both serial and parallel processing. These node types can include thin nodes, wide nodes, 604 high nodes, as well as other types of nodes both internal and external to the SP frame.

Node Switch Board. Switches mounted on frames that contain nodes.

NSB. Node Switch Board.

NTP. Network Time Protocol.

O

ODM. Object Data Manager. In AIX, a hierarchical object-oriented database for configuration data.

P

parallel environment. A system environment where message passing or SP resource manager services are used by the application.

Parallel Environment. A licensed IBM program used for message passing applications on the SP or RS/6000 platforms.

parallel processing. A multiprocessor architecture which allows processes to be allocated to tightly coupled multiple processors in a cooperative processing environment, allowing concurrent execution of tasks.

parameter. * (1) A variable that is given a constant value for a specified application and that may denote the application. * (2) An item in a menu for which the operator specifies a value or for which the system provides a value when the menu is interpreted. * (3) A name in a procedure that is used to refer to an argument that is passed to the procedure. * (4) A particular piece of information that a system or application program needs to process a request.

partition. See system partition.

Perl. Practical Extraction and Report Language.

perspective. The primary window for each SP Perspectives application, so called because it provides a unique view of an SP system.

pipe. A UNIX utility allowing the output of one command to be the input of another. Represented by the | symbol. It is also referred to as filtering output.

PMR. Problem Management Report.

POE. Formerly Parallel Operating Environment, now Parallel Environment for AIX.

port. (1) An end point for communication between devices, generally referring to physical connection. (2) A 16-bit number identifying a particular TCP or UDP resource within a given TCP/IP node.

predicate. Obsolete term for expression.

Primary node or machine. (1) A device that runs a workload and has a standby device ready to assume the primary workload if that primary node fails or is taken out of service. (2) A node on the switch that initializes, provides diagnosis and recovery services, and performs other operations to the switch network. (3) In IBM Virtual Shared Disk function, when physical disks are connected to two nodes (twin-tailed), one node is designated as the primary node for each disk and the other is designated the secondary, or backup, node. The primary node is the server node for IBM Virtual Shared Disks defined on the physical disks under normal conditions. The secondary node can become the server node for the disks if the primary node is unavailable (off-line or down).

Problem Management Report. The number in the IBM support mechanism that represents a service incident with a customer.

process. * (1) A unique, finite course of events defined by its purpose or by its effect, achieved under defined conditions. * (2) Any operation or combination of operations on data. * (3) A function being performed or waiting to be performed. * (4) A program in operation. For example, a daemon is a system process that is always running on the system.

protocol. A set of semantic and syntactic rules that defines the behavior of functional units in achieving communication.

R

RAID. Redundant array of independent disks.

rearm expression. In Event Management, an expression used to generate an event that alternates with an original event expression in the following way: the event expression is used until it is true, then the rearm expression is used until it is true, then the event expression is used, and so on. The rearm expression is commonly the inverse of the event expression (for example, a resource variable is on or off). It can also be used with the event expression to define an upper and lower boundary for a condition of interest.

rearm predicate. Obsolete term for rearm expression.

remote host. *See foreign host*.

resource. In Event Management, an entity in the system that provides a set of services. Examples of resources include hardware entities such as processors, disk drives, memory, and adapters, and software entities such as database applications, processes, and file systems. Each resource in the system has one or more attributes that define the state of the resource.

resource identifier. In Event Management, a set of elements, where each element is a name/value pair of the form `name=value`, whose values uniquely identify the copy of the resource (and by extension, the copy of the resource variable) in the system.

resource monitor. A program that supplies information about resources in the system. It can be a command, a daemon, or part of an application or subsystem that manages any type of system resource.

resource variable. In Event Management, the representation of an attribute of a resource. An example of a resource variable is `IBM.AIX.PagSp.%totalfree`, which represents the percentage of total free paging space. `IBM.AIX.PagSp` specifies the resource name and `%totalfree` specifies the resource attribute.

RISC. Reduced Instruction Set Computing (RISC), the technology for today's high performance personal computers and workstations, was invented in 1975. Uses a small simplified set of frequently used instructions for rapid execution.

rlogin (remote LOGIN). A service offered by Berkeley UNIX systems that allows authorized users of one machine to connect to other UNIX systems across a network and interact as if their terminals were connected directly. The rlogin software passes information about the user's environment (for example, terminal type) to the remote machine.

RPC. Acronym for Remote Procedure Call, a facility that a client uses to have a server execute a procedure call. This facility is composed of a library of procedures plus an XDR.

RSH. A variant of RLOGIN command that invokes a command interpreter on a remote UNIX machine and passes the command line arguments to the command interpreter, skipping the LOGIN step completely. See also *rlogin*.

S

SCSI. Small Computer System Interface.

Secondary node. In IBM Virtual Shared Disk function, when physical disks are connected to two nodes (twin-tailed), one node is designated as the primary node for each disk and the other is designated as the secondary, or backup, node. The secondary node acts as the server node for the IBM Virtual Shared disks defined on the physical disks if the primary node is unavailable (off-line or down).

server. (1) A function that provides services for users. A machine may run client and server processes at the same time. (2) A machine that provides resources to the network. It provides a network service, such as disk storage and file transfer, or a program that uses such a service. (3) A device, program, or code module on a network dedicated to providing a specific service to a network. (4) On a LAN, a data station that provides facilities to other data stations. Examples are file server, print server, and mail server.

shell. The shell is the primary user interface for the UNIX operating system. It serves as command language interpreter, programming language, and allows foreground and background processing. There are three different implementations of the shell concept: Bourne, C and Korn.

Small Computer System Interface (SCSI). An input and output bus that provides a standard interface for the attachment of various direct access storage devices (DASD) and tape drives to the RS/6000.

Small Computer Systems Interface Adapter (SCSI Adapter). An adapter that supports the attachment of various direct-access storage devices (DASD) and tape drives to the RS/6000.

SMIT. The System Management Interface Toolkit is a set of menu driven utilities for AIX that provides functions such as transaction login, shell script creation, automatic updates of object database, and so forth.

SNMP. Simple Network Management Protocol. (1) An IP network management protocol that is used to monitor attached networks and routers. (2) A TCP/IP-based protocol for exchanging network management information and outlining the structure for communications among network devices.

socket. (1) An abstraction used by Berkeley UNIX that allows an application to access TCP/IP protocol functions. (2) An IP address and port number pairing. (3) In TCP/IP, the Internet address of the host computer on which the application runs, and the port number it uses. A TCP/IP application is identified by its socket.

standby node or machine. A device that waits for a failure of a primary node in order to assume the identity of the primary node. The standby machine then runs the primary's workload until the primary is back in service.

subnet. Shortened form of subnetwork.

subnet mask. A bit template that identifies to the TCP/IP protocol code the bits of the host address that are to be used for routing for specific subnetworks.

subnetwork. Any group of nodes that have a set of common characteristics, such as the same network ID.

subsystem. A software component that is not usually associated with a user command. It is usually a daemon process. A subsystem will perform work or provide services on behalf of a user request or operating system request.

SUP. Software Update Protocol.

switch capsule. A group of SP frames consisting of a switched frame and its companion non-switched frames.

Sysctl. Secure System Command Execution Tool. An authenticated client/server system for running commands remotely and in parallel.

syslog. A BSD logging system used to collect and manage other subsystem's logging data.

System Administrator. The user who is responsible for setting up, modifying, and maintaining the SP system.

system partition. A group of nonoverlapping nodes on a switch chip boundary that act as a logical SP system.

T

tar. Tape ARchive, is a standard UNIX data archive utility for storing data on tape media.

TaskGuides. SP TaskGuides are a form of advanced online assistance designed to walk you through complex or infrequently performed tasks. Each TaskGuide does not simply list the required steps. It actually performs the steps for you, automating the steps to the highest degree possible and prompting you for input only when absolutely necessary. You might recognize them as *wizards*.

Tcl. Tool Command Language.

TclX. Tool Command Language Extended.

TCP. Acronym for Transmission Control Protocol, a stream communication protocol that includes error recovery and flow control.

TCP/IP. Acronym for Transmission Control Protocol/Internet Protocol, a suite of protocols designed to allow communication between networks regardless of the technologies implemented in each network. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It assumes that the underlying protocol is the Internet Protocol.

Telnet. Terminal Emulation Protocol, a TCP/IP application protocol that allows interactive access to foreign hosts.

| **ticket**. An encrypted protocol message used to
| securely pass the identity of a user from a client to a
| server.

Tk. Tcl-based Tool Kit for X Windows.

TMPCP. Tape Management Program Control Point.

token-ring. (1) Network technology that controls media access by passing a token (special packet or frame) between media-attached machines. (2) A network with a ring topology that passes tokens from one attaching device (node) to another. (3) The IBM Token-Ring LAN connection allows the RS/6000 system unit to participate in a LAN adhering to the IEEE 802.5 Token-Passing Ring standard or the ECMA standard 89 for Token-Ring, baseband LANs.

transaction. An exchange between the user and the system. Each activity the system performs for the user is considered a transaction.

transceiver (transmitter-receiver). A physical device that connects a host interface to a local area network, such as Ethernet. Ethernet transceivers contain electronics that apply signals to the cable and sense collisions.

transfer. To send data from one place and to receive the data at another place. Synonymous with move.

transmission. * The sending of data from one place for reception elsewhere.

TURBOWAYS 100 ATM Adapter. An IBM high-performance, high-function intelligent adapter that provides dedicated 100 Mbps ATM (asynchronous transfer mode) connection for high-performance servers and workstations.

U

UDP. User Datagram Protocol.

UNIX operating system. An operating system developed by Bell Laboratories that features multiprogramming in a multiuser environment. The UNIX operating system was originally developed for use on minicomputers, but has been adapted for mainframes and microcomputers. **Note:** The AIX operating system is IBM's implementation of the UNIX operating system.

user. Anyone who requires the services of a computing system.

User Datagram Protocol (UDP). (1) In TCP/IP, a packet-level protocol built directly on the Internet Protocol layer. UDP is used for application-to-application programs between TCP/IP host systems. (2) A transport protocol in the Internet suite of protocols that provides unreliable, connectionless datagram service. (3) The Internet Protocol that enables an application programmer on one machine or process to send a datagram to an application program on another machine or process.

user ID. A nonnegative integer, contained in an object of type *uid_t*, that is used to uniquely identify a system user.

V

Virtual Shared Disk, IBM. The function that allows application programs executing at different nodes of a system partition to access a raw logical volume as if it were local at each of the nodes. In actuality, the logical volume is local at only one of the nodes (the server node).

W

workstation. * (1) A configuration of input/output equipment at which an operator works. * (2) A terminal or microcomputer, usually one that is connected to a mainframe or to a network, at which a user can perform applications.

X

X Window System. A graphical user interface product.

Bibliography

This bibliography helps you find product documentation related to the RS/6000 SP hardware and software products.

You can find most of the IBM product information for RS/6000 SP products on the World Wide Web. Formats for both viewing and downloading are available.

PSSP documentation is shipped with the PSSP product in a variety of formats and can be installed on your system. The man pages for public code that PSSP includes are also available online.

You can order hard copies of the product documentation from IBM. This bibliography lists the titles that are available and their order numbers.

Finally, this bibliography contains a list of non-IBM publications that discuss parallel computing and other topics related to the RS/6000 SP.

Information Formats

Documentation supporting RS/6000 SP software licensed programs is no longer available from IBM in hardcopy format. However, you can view, search, and print documentation in the following ways:

- On the World Wide Web
- Online (from the product media or the SP Resource Center)

Finding Documentation on the World Wide Web

Most of the RS/6000 SP hardware and software books are available from the IBM RS/6000 Web site at:

http://www.rs6000.ibm.com

You can view a book or download a Portable Document Format (PDF) version of it. At the time this manual was published, the Web address of the "RS/6000 SP Product Documentation Library" page was:

http://www.rs6000.ibm.com/resource/aix_resource/sp_books

However, the structure of the RS/6000 Web site can change over time.

Accessing PSSP Documentation Online

On the same medium as the PSSP product code, IBM ships PSSP man pages, HTML files, and PDF files. In order to use these publications, you must first install the **ssp.docs** file set.

To view the PSSP HTML publications, you need access to an HTML document browser such as Netscape. The HTML files and an index that links to them are installed in the **/usr/lpp/ssp/html** directory. Once installed, you can also view the HTML files from the RS/6000 SP Resource Center.

If you have installed the SP Resource Center on your SP system, you can access it by entering the **/usr/lpp/ssp/bin/resource_center** command. If you have the SP Resource Center on CD-ROM, see the **readme.txt** file for information about how to run it.

To view the PSSP PDF publications, you need access to the Adobe Acrobat Reader. The Acrobat Reader is shipped with the AIX Version 4.3 Bonus Pack and is also freely available for downloading from the Adobe Web site at:

http://www.adobe.com

To successfully print a large PDF file (approximately 300 or more pages) from the Adobe Acrobat reader, you may need to select the “Download Fonts Once” button on the Print window.

Manual Pages for Public Code

The following manual pages for public code are available in this product:

SUP /usr/lpp/ssp/man/man1/sup.1

Perl (Version 4.036) /usr/lpp/ssp/perl/man/perl.man

/usr/lpp/ssp/perl/man/h2ph.man

/usr/lpp/ssp/perl/man/s2p.man

/usr/lpp/ssp/perl/man/a2p.man

Manual pages and other documentation for **Tcl**, **TclX**, **Tk**, and **expect** can be found in the compressed **tar** files located in the **/usr/lpp/ssp/public** directory.

RS/6000 SP Planning Publications

This section lists the IBM product documentation for planning for the IBM RS/6000 SP hardware and software.

IBM RS/6000 SP:

- *Planning, Volume 1, Hardware*

  and Physical Environment, GA22-7280
- *Planning, Volume 2, Control*

  Workstation and Software Environment, GA22-7281

RS/6000 SP Hardware Publications

This section lists the IBM product documentation for the IBM RS/6000 SP hardware.

IBM RS/6000 SP:

- *Planning, Volume 1, Hardware*

  and Physical Environment, GA22-7280
- *Planning, Volume 2, Control*

  Workstation and Software Environment, GA22-7281
- *Installation and*

  Relocation, GA22-7441
- *System Service Guide*, GA22-7442
- *SP Switch Service*

Guide, GA22-7443

- *SP Switch2 Service Guide*, GA22-7444
- *Uniprocessor Node Service Guide*, GA22-7445
- *604 and 604e SMP High Node Service Guide*, GA22-7446
- *SMP Thin and Wide Node Service Guide*, GA22-7447
- *POWER3 SMP High Node Service Guide*, GA22-7448

RS/6000 SP Switch Router Publications

The RS/6000 SP Switch Router is based on the Ascend GRF switched IP router product from Lucent Technologies. You can order the SP Switch Router as the IBM 9077.

The following publications are shipped with the SP Switch Router. You can also order these publications from IBM using the order numbers shown.

- *Ascend GRF GateD Manual*, GA22-7327
- *Ascend GRF 400/1600 Getting Started*, GA22-7368
- *Ascend GRF Configuration and Management*, GA22-7366
- *Ascend GRF Reference Guide*, GA22-7367
- *SP Switch Router Adapter Guide*, GA22-7310

RS/6000 SP Software Publications

This section lists the IBM product documentation for software products related to the IBM RS/6000 SP. These products include:

- IBM Parallel System Support Programs for AIX (PSSP)
- IBM LoadLeveler for AIX (LoadLeveler)
- IBM Parallel Environment for AIX (Parallel Environment)
- IBM General Parallel File System for AIX (GPFS)
- IBM Engineering and Scientific Subroutine Library (ESSL) for AIX
- IBM Parallel ESSL for AIX
- IBM High Availability Cluster Multi-Processing for AIX (HACMP)
- IBM Client Input Output/Sockets (CLIO/S)

- IBM Network Tape Access and Control System for AIX (NetTAPE)

PSSP Publications

IBM RS/6000 SP:

- *Planning, Volume 2, Control*

  Workstation and Software Environment, GA22-7281

PSSP:

- *Installation and Migration*

  Guide, GA22-7347
- *Administration Guide*, SA22-7348
- *Managing Shared Disks*, SA22-7349
- *Performance Monitoring Guide*

  and Reference, SA22-7353
- *Diagnosis Guide*, GA22-7350
- *Command and Technical Reference*, SA22-7351
- *Messages Reference*, GA22-7352

RS/6000 Cluster Technology

(RSCT):

- *Event Management Programming*

  Guide and Reference, SA22-7354
- *Group Services Programming*

  Guide and Reference, SA22-7355
- *First Failure Data*

  Capture Programming Guide and Reference, SA22-7454

LoadLeveler Publications

LoadLeveler:

- *Using and Administering*, SA22-7311
- *Diagnosis and Messages Guide*, GA22-7277

GPFS Publications

GPFS:

- *Problem Determination*

  Guide, GA22-7434
- *Data Management*

  API Guide, GA22-7435
- *Guide and Reference*, GA22-7452
- *Installation and*

| *Tuning Guide*, GA22-7453

Parallel Environment Publications

Parallel Environment:

| • *Installation Guide*, GA22-7418
| • *Messages*, GA22-7419
| • *DPCL Programming Guide*, SA22-7420
| • *DPCL Class Reference*, SA22-7421
| • *MPI Programming Guide*, SA22-7422
| • *MPI Subroutine Reference*, SA22-7423
| • *Hitchhiker's Guide*, SA22-7424
| • *Operation and Use, Volume*
| *1*, SA22-7425
| • *Operation and Use, Volume*
| *2*, SA22-7426
| • *MPL Programming*
| *and Subroutine Reference*, GC23-3893

Parallel ESSL and ESSL Publications

- *ESSL Products: General Information*, GC23-0529
- *Parallel ESSL: Guide and*
  Reference, SA22-7273
- *ESSL: Guide and Reference*, SA22-7272

HACMP Publications

HACMP:

- *Concepts and Facilities*, SC23-4276
- *Planning Guide*, SC23-4277
- *Installation Guide*, SC23-4278
- *Administration Guide*, SC23-4279
- *Troubleshooting Guide*, SC23-4280
- *Programming Locking Applications*, SC23-4281
- *Programming Client Applications*, SC23-4282
- *Master Index and Glossary*, SC23-4285
- *HANFS for AIX Installation*
  and Administration Guide, SC23-4283
- *Enhanced Scalability Installation*
  and Administration Guide, SC23-4284

CLIO/S Publications

CLIO/S:

- *General Information*, GC23-3879
- *User's Guide and Reference*, GC28-1676

NetTAPE Publications

NetTAPE:

- *General Information*, GC23-3990
- *User's Guide and Reference*, available from your IBM representative

AIX and Related Product Publications

For the latest information on AIX and related products, including RS/6000 hardware products, see *AIX and*

Related Products Documentation Overview, SC23-2456. You can order a hard copy of the book from IBM. You can also view it online from the “AIX Online Publications and Books” page of the RS/6000 Web site at:

http://www.rs6000.ibm.com/resource/aix_resource/Pubs

DCE Publications

The DCE library consists of the following books:

- *IBM DCE 3.1 for AIX: Administration*

  Commands Reference
- *IBM DCE 3.1 for*

  AIX: Administration Guide—Introduction
- *IBM DCE 3.1 for*

  AIX: Administration Guide—Core Components
- *IBM DCE 3.1 for AIX: DFS*

  Administration Guide and Reference
- *IBM DCE 3.1 for*

  AIX: Application Development Guide—Introduction and Style Guide
- *IBM DCE 3.1 for*

  AIX: Application Development Guide—Core Components
- *IBM DCE 3.1 for*

  AIX: Application Development Guide—Directory Services
- *IBM DCE 3.1 for AIX: Application*

  Development Reference
- *IBM DCE 3.1 for AIX: Problem*

  Determination Guide
- *IBM DCE 3.1 for AIX: Release*

Notes

You can view a DCE book or download a Portable Document Format (PDF) version of it from the IBM DCE Web site at:

http://www.ibm.com/software/network/dce/library

Red Books

IBM's International Technical Support Organization (ITSO) has published a number of redbooks related to the RS/6000 SP. For a current list, see the ITSO Web site at:

http://www.redbooks.ibm.com

Non-IBM Publications

Here are some non-IBM publications that you may find helpful.

- Almasi, G., Gottlieb, A., *Highly Parallel Computing*, Benjamin-Cummings Publishing Company, Inc., 1989.
- Foster, I., *Designing and Building Parallel Programs*, Addison-Wesley, 1995.
- Gropp, W., Lusk, E., Skjellum, A., *Using MPI*, The MIT Press, 1994.
- Message Passing Interface Forum, *MPI: A Message-Passing Interface Standard, Version 1.1*, University of Tennessee, Knoxville, Tennessee, June 6, 1995.
- Message Passing Interface Forum, *MPI-2: Extensions to the Message-Passing Interface, Version 2.0*, University of Tennessee, Knoxville, Tennessee, July 18, 1997.
- Ousterhout, John K., *Tcl and the Tk Toolkit*, Addison-Wesley, Reading, MA, 1994, ISBN 0-201-63337-X.
- Pfister, Gregory, F., *In Search of Clusters*, Prentice Hall, 1998.

Index

D

Communicating Your Comments to IBM

Parallel System Support Programs for AIX
Command and Technical
Reference, Volume 1
Version 3 Release 2

Publication No. SA22-7351-02

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a reader's comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:
  - FAX: (International Access Code)+1+914+432-9405
- If you prefer to send comments electronically, use one of these network IDs:
  - IBM Mail Exchange: USIB6TC9 at IBMMAIL
  - Internet e-mail: mhvrcfs@us.ibm.com

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies

Optionally, if you include your telephone number, we will be able to respond to your comments by phone.

Reader's Comments — We'd Like to Hear from You

Parallel System Support Programs for AIX
Command and Technical
Reference, Volume 1
Version 3 Release 2

Publication No. SA22-7351-02

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Today's date: ____________________________________

What is your occupation?

Newsletter number of latest Technical Newsletter (if any) concerning this publication:

How did you use this publication?

[ ]	As an introduction	[ ]	As a text (student)
[ ]	As a reference manual	[ ]	As a text (instructor)
[ ]	For another purpose (explain)		

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number: Comment:

Name

Address

Company or Organization

Phone No.

Cut or Fold
Along Line

Reader's Comments — We'd Like to Hear from You
SA22-7351-02

IBM®

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie NY 12601-5400

Fold and Tape **Please do not staple** Fold and Tape

SA22-7351-02

Cut or Fold
Along Line

Program Number: 5765-D51

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SA22-7351-02