Process and Memory Management Commands on Cisco IOS XR Software

This chapter describes the Cisco IOS XR software commands used to manage processes and memory.

For more information about using the process and memory management commands to perform troubleshooting tasks, see *Cisco IOS XR Getting Started Guide*.

affinity location-set

To set the affinity of a placeable program (process) to or from a group of nodes, use the **affinity location-set** command in placement program configuration mode. To remove the affinity location set, use the **no** form of this command.

affinity location-set *node-id1* [...[*node-id5*]] {**attract** *strength* | **repulse** *strength* | **default** | **none**}

no affinity location-set *node-id1* [...[*node-id5*]] {**attract** *strength* | **repulse** *strength* | **default** | **none**}

Syntax Description

[...[*node-id5*]]	The *node-id* argument is expressed in the *rack*/*slot*/*module* notation. To specify multiple nodes, enter the *node-id* for each node. You can specify a node pair, or up to 5 nodes.
attract	Specifies a positive affinity.
repulse	Specifies a negative affinity.
strength	A positive or negative strength. The range is from 1 to 100000 points.
default	Sets the affinity location-set default value, which is equivalent to specifying an **attract** *strength* of 200.
none	Sets the affinity value to zero points.

Defaults

There are no default location-set affinities for any placeable processes.
Specifying an affinity value of **default** is equivalent to specifying the **attract** *strength* argument to be 200.

Command Modes

Placement program configuration

Command History

Release	Modification
Release 3.3.0	This command was introduced on the Cisco CRS-1.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

The **affinity location-set** command assigns an affinity for a process to run on a specific node, node pair or set of up to 5 nodes.

A node pair is either an active and standby pair of nodes [hosted on route processors (RPs) or distributed RPs], or a single active node on an RP or DRP that does not have a standby.

By containing references to specific location identifiers, location-set affinities are more specific to the individual SDR in which they are used than other affinity types.

Task ID

Task ID	Operations
sysmgr	read, write

Examples

The following example shows how to configure placement affinity for the OSPF process class on the node pair:

```
RP/0/RP0/CPU0:router(config)# placement program ospf
RP/0/RP0/CPU0:router(config-place)# affinity location-set 0/1/cpu0 0/1/cpu1 attract 200
```

The following example shows how to stop loading a DRP with more processes:

```
RP/0/RP0/CPU0:router(config)# placement program default
RP/0/RP0/CPU0:router(config-place)# affinity location-set 0/1/cpu0 repulse 90
```

Related Commands

Command	Description
placement program	Enters placement program configuration mode to set process affinities (preferences).

affinity location-type

To set the affinity of a placeable program (process) to or from a location type, use the **affinity location-set** command in placement program configuration mode. To remove the affinity of a location set, use the **no** form of this command.

affinity location-type {**current** | **paired** | **primary**} {**attract** *strength* | **repulse** *strength* | **default** | **none**}

no affinity location-type {**current** | **paired** | **primary**} {**attract** *strength* | **repulse** *strength* | **default** | **none**}‘

Syntax Description

current	Affinity for the current location.
paired	Affinity for a node (or route processor) with a standby node.
primary	Affinity for the primary node.
attract	Specifies a positive affinity.
repulse	Specifies a negative affinity.
strength	A positive or negative strength. The range is from 1 to 100000 points.
default	Sets the affinity location-type default values.
none	Sets the affinity value to zero points.

Defaults

By default, all processes have the following settings:

- **location-type current attract** *strength* argument: 100
- **location-type paired attract** *strength* argument: 60
- **location-type primary attract** *strength* argument: 40

Specifying an affinity value of **default** is equivalent to specifying the **attract** *strength* argument to be 200.

Command Modes

Placement program configuration

Command History

Release	Modification
Release 3.2.50	This command was introduced on the Cisco CRS-1.
Release 3.3.0	No modification.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

The **affinity location-type** command is used to place a process at a particular location. The default policy is that the location type be a node pair (active and standby nodes), and that the process does not move automatically (**current** = 100) unless a solo node fails and the process must be started on a different node.

You can configure the placement policy to allow certain processes to stay where they are (**current**) or move just by indicating so through the various affinity choices. The higher the positive value of an affinity, the stronger the requirement that the process run at a location, and so on. A low or zero point value would indicate a weaker requirement (or no preference) that a process run at a location.

The affinity location type with the **paired** option configures the affinity towards the nodes that are paired (active and standby nodes), and as mentioned earlier, this is the default policy. The affinity location type using the **primary** option configures the affinity towards the designated secure domain router system controller (DSDRSC) node.

Note Change the **current** affinity with assistance from Cisco support personnel.

Task ID

Task ID	Operations
sysmgr	read, write

Examples

This example shows how to place Border Gateway Protocol (BGP) in a specific location by setting it with a high attract value. The result is that BGP does not move location under any circumstances other than a node pair removal.

```
RP/0/RP0/CPU0:router(config)# placement program bgp
RP/0/RP0/CPU0:router(config-place)# affinity location-type current attract 5000
```

Related Commands

Command	Description
placement program	Enters placement program configuration mode to set process affinities (preferences).

affinity program

To set the affinity of a placeable program (process) to or from another program, use the **affinity program** command in placement program configuration mode. To remove the affinity of a program, use the **no** form of this command.

affinity program *program* {**attract** *strength* | **repulse** *strength* | **default** | **none**}

no affinity program *program* {**attract** *strength* | **repulse** *strength* | **default** | **none**}

Syntax Description

program	Name of specific a program or program group.
attract	Specifies a positive affinity.
repulse	Specifies a negative affinity.
strength	A positive or negative strength. The range is from 1 to 100000.
default	Sets the affinity value to the default.
none	Sets the affinity value to zero.

Defaults

Specifying an affinity value of **default** is equivalent to specifying the **attract** *strength* argument to be 200.

Command Modes

Placement program configuration

Command History

Release	Modification
Release 3.3.0	This command was introduced on the Cisco CRS-1.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

The **affinity program** command is used to collocate processes or keep them apart. You would want to use this command because you have learned that certain processes perform better when they are running together on the same node (**attract**)—or on different nodes, apart from each other (**repulse**).

Task ID

Task ID	Operations
sysmgr	read, write

Examples

The following example shows how to keep IPv4 and IPv6 Routing Information Bases (RIBs) apart:

```
RP/0/RP0/CPU0:router(config)# placement program ipv4_rib
RP/0/RP0/CPU0:router(config-place)# affinity program ipv6_rib repulse 200
```

Related Commands

Command	Description
placement program	Enters placement program configuration mode to set process affinities (preferences).

affinity self

To set the affinity of a placeable program (process) to or from one of its own instances, use the **affinity self** command in placement program configuration mode. To remove the affinity self setting, use the **no** form of this command.

affinity self {**attract** *strength* | **repulse** *strength* | **default** | **none**}

no affinity self {**attract** *strength* | **repulse** *strength* | **default** | **none**}

Syntax Description

attract	Specifies a positive affinity.
repulse	Specifies a negative affinity.
strength	A positive or negative strength. The range is from 1 to 100000 points.
default	Sets the affinity value to the default.
none	Sets the affinity value to zero points.

Defaults

The affinity self default is initially set in system placement files.

Command Modes

Placement program configuration

Command History

Release	Modification
Release 3.3.0	This command was introduced on the Cisco CRS-1.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

The **affinity self** command is used to adjust placement decisions when multiple instances of a process are started. An attract (positive) affinity indicates a preference to have all instances of a process run on the same node, while a repulse (negative) affinity indicates a preference to have each instance of a process run on different nodes.

Typically, you would want to monitor how a process on your system best performs, and whether instances of the same class should be collocated or not.

Task ID

Task ID	Operations
sysmgr	read, write

Examples

The following example shows how to configure BGP speakers to run on different nodes:

```
RP/0/RP0/CPU0:router(config)# placement program bgp
RP/0/RP0/CPU0:router(config-place)# affinity self repulse 200
```

Related Commands

Command	Description
placement program	Enters placement program configuration mode to set process affinities (preferences).

clear context

To clear core dump context information, use the **clear context** command in administration EXEC mode or in EXEC mode.

clear context location {*node-id* | **all**}

Syntax Description

location *node-id*	(Optional) Clears core dump context information for a specified node. The *node-id* argument is expressed in the *rack*/*slot*/*module* notation.
location all	(Optional) Clears core dump context information for all nodes.

Defaults

No default behavior or values

Command Modes

Administration EXEC
EXEC

Command History

Release	Modification
Release 2.0	This command was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	This command was first supported on the Cisco XR 12000 Series Router. The command was made available in administration EXEC mode.
Release 3.3.0	No modification.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

Use the **clear context** command to clear core dump context information. If you do not specify a node with the **location** keyword and *node-id* argument, this command clears core dump context information for all nodes.

Use the **show context** command to display core dump context information.

Task ID

Task ID	Operations
diag	execute

Examples

The following example shows how to clear core dump context information:

```
RP/0/RP0/CPU0:router# clear context
```

Related Commands

Command	Description
show context	Displays core dump context information.

dumpcore

To manually generate a core dump, use the **dumpcore** command in administration EXEC mode or in EXEC mode.

dumpcore {**running** | **suspended**} *job-id* **location** *node-id*

Syntax Description

running	Generates a core dump for a running process.
suspended	Suspends a process, generates a core dump for the process, and resumes the process.
job-id	Process instance identifier.
location *node-id*	Generates a core dump for a process running on the specified node. The *node-id* argument is expressed in the *rack*/*slot*/*module* notation.

Defaults

No default behavior or values

Command Modes

Administration EXEC
EXEC

Command History

Release	Modification
Release 2.0	This command was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	This command was first supported on the Cisco XR 12000 Series Router. The command was made available in administration EXEC mode.
Release 3.3.0	No modification.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

When a process crashes on the Cisco IOS XR software, a core dump file of the event is written to a designated destination without bringing down the router. Upon receiving notification that a process has terminated abnormally, the Cisco IOS XR software then respawns the crashed process. Core dump files are used by Cisco Technical Support Center engineers and development engineers to debug the Cisco IOS XR software.

On the Cisco IOS XR software, core dumps can be generated manually for a process, even when a process has not crashed. Two modes exist to generate a core dump manually:

- **running**—Generates a core dump for a running process. This mode can be used to generate a core dump on a critical process (a process whose suspension could have a negative impact on the performance of the router) because the core dump file is generated independently, that is, the process continues to run as the core dump file is being generated.
- **suspended**—Suspends a process, generates a core dump for the process, and resumes the process. Because the process is suspended, this mode ensures data consistency in the core dump file.

Core dump files contain the following information about a crashed process:

- Register information
- Thread status information
- Process status information
- Selected memory segments

Task ID

Task ID	Operations
diag	read, write

Examples

The following example shows how to generate a core dump in suspended mode for the process instance 52:

```
RP/0/RP0/CPU0:router# dumpcore suspended 52

RP/0/RP0/CPU0:Sep 22 01:40:26.982 : sysmgr[71]: process in stop/continue state 4104
RP/0/RP0/CPU0Sep 22 01:40:26.989 : dumper[54]: %DUMPER-4-CORE_INFO : Core for pid = 4104
(pkg/bin/devc-conaux) requested by pkg/bin/dumper_gen@node0_RP0_CPU0
RP/0/RP0/CPU0Sep 22 01:40:26.993 : dumper[54]: %DUMPER-6-SPARSE_CORE_DUMP : Sparse core
dump as configured dump sparse for all
RP/0/RP0/CPU0Sep 22 01:40:26.995 : dumper[54]: %DUMPER-7-DLL_INFO_HEAD : DLL path    Text
addr.  Text size   Data addr.  Data size   Version
RP/0/RP0/CPU0Sep 22 01:40:26.996 : dumper[54]: %DUMPER-7-DLL_INFO :
/pkg/lib/libplatform.dll  0xfc0d5000  0x0000a914  0xfc0e0000  0x00002000        0
RP/0/RP0/CPU0Sep 22 01:40:26.996 : dumper[54]: %DUMPER-7-DLL_INFO : /pkg/lib/libsysmgr.dll
0xfc0e2000  0x0000ab48  0xfc0c295c  0x00000368        0
RP/0/RP0/CPU0Sep 22 01:40:26.997 : dumper[54]: %DUMPER-7-DLL_INFO : /pkg/lib/libinfra.dll
0xfc0ed000  0x00032de0  0xfc120000  0x00000c90        0
RP/0/RP0/CPU0Sep 22 01:40:26.997 : dumper[54]: %DUMPER-7-DLL_INFO : /pkg/lib/libios.dll
0xfc121000  0x0002c4bc  0xfc14e000  0x00002000        0
RP/0/RP0/CPU0Sep 22 01:40:26.997 : dumper[54]: %DUMPER-7-DLL_INFO : /pkg/lib/libc.dll
0xfc150000  0x00077ae0  0xfc1c8000  0x00002000        0
RP/0/RP0/CPU0Sep 22 01:40:26.998 : dumper[54]: %DUMPER-7-DLL_INFO : /pkg/lib/libsyslog.dll
0xfc1d2000  0x0000530c  0xfc120c90  0x00000308        0
RP/0/RP0/CPU0Sep 22 01:40:26.998 : dumper[54]: %DUMPER-7-DLL_INFO :
/pkg/lib/libbackplane.dll  0xfc1d8000  0x0000134c  0xfc0c2e4c  0x000000a8        0
RP/0/RP0/CPU0Sep 22 01:40:26.999 : dumper[54]: %DUMPER-7-DLL_INFO : /pkg/lib/libnodeid.dll
0xfc1e5000  0x00009114  0xfc1e41a8  0x00000208        0
RP/0/RP0/CPU0Sep 22 01:40:26.999 : dumper[54]: %DUMPER-7-DLL_INFO :
/pkg/lib/libttyserver.dll  0xfc1f1000  0x0003dfcc  0xfc22f000  0x00002000        0
RP/0/RP0/CPU0Sep 22 01:40:27.000 : dumper[54]: %DUMPER-7-DLL_INFO :
/pkg/lib/libttytrace.dll  0xfc236000  0x00004024  0xfc1e44b8  0x000001c8        0
RP/0/RP0/CPU0Sep 22 01:40:27.000 : dumper[54]: %DUMPER-7-DLL_INFO : /pkg/lib/libdebug.dll
0xfc23b000  0x0000ef64  0xfc1e4680  0x00000550        0
RP/0/RP0/CPU0Sep 22 01:40:27.001 : dumper[54]: %DUMPER-7-DLL_INFO :
/pkg/lib/lib_procfs_util.dll  0xfc24a000  0x00004e2c  0xfc1e4bd0  0x000002a8        0
```

```
RP/0/RP0/CPU0Sep 22 01:40:27.001 : dumper[54]: %DUMPER-7-DLL_INFO : /pkg/lib/libsysdb.dll
0xfc24f000  0x000452e0  0xfc295000  0x00000758        0
RP/0/RP0/CPU0:Oakland#RP/0/RP0/CPU0Sep 22 01:40:27.001 : dumper[54]: %DUMPER-7-DLL_INFO :
/pkg/lib/libsysdbutils.dll  0xfc296000  0x0000ae08  0xfc295758  0x000003ec        0
RP/0/RP0/CPU0Sep 22 01:40:27.002 : dumper[54]: %DUMPER-7-DLL_INFO :
/pkg/lib/lib_tty_svr_error.dll  0xfc2a1000  0x0000172c  0xfc1e4e78  0x00000088        0
RP/0/RP0/CPU0Sep 22 01:40:27.002 : dumper[54]: %DUMPER-7-DLL_INFO :
/pkg/lib/lib_tty_error.dll  0xfc2a3000  0x00001610  0xfc1e4f00  0x00000088        0
RP/0/RP0/CPU0Sep 22 01:40:27.003 : dumper[54]: %DUMPER-7-DLL_INFO : /pkg/lib/libwd_evm.dll
0xfc2a5000  0x0000481c  0xfc295b44  0x00000188        0
RP/0/RP0/CPU0Sep 22 01:40:27.003 : dumper[54]: %DUMPER-7-DLL_INFO : /pkg/lib/libttydb.dll
0xfc2aa000  0x000051dc  0xfc295ccc  0x00000188        0
RP/0/RP0/CPU0Sep 22 01:40:27.004 : dumper[54]: %DUMPER-7-DLL_INFO :
/pkg/lib/libttydb_error.dll  0xfc23a024  0x00000f0c  0xfc295e54  0x00000088        0
RP/0/RP0/CPU0Sep 22 01:40:27.004 : dumper[54]: %DUMPER-7-DLL_INFO : /pkg/lib/librs232.dll
0xfc2b0000  0x00009c28  0xfc2ba000  0x00000470        0
RP/0/RP0/CPU0Sep 22 01:40:27.005 : dumper[54]: %DUMPER-7-DLL_INFO :
/pkg/lib/lib_rs232_error.dll  0xfc2bb000  0x00000f8c  0xfc295edc  0x00000088        0
RP/0/RP0/CPU0Sep 22 01:40:27.005 : dumper[54]: %DUMPER-7-DLL_INFO :
/pkg/lib/libst16550.dll  0xfc2bc000  0x00008ed4  0xfc2ba470  0x00000430        0
RP/0/RP0/CPU0Sep 22 01:40:27.006 : dumper[54]: %DUMPER-7-DLL_INFO : /pkg/lib/libconaux.dll
0xfc2c5000  0x00001dc0  0xfc2ba8a0  0x000001a8        0
RP/0/RP0/CPU0Sep 22 01:40:27.006 : dumper[54]: %DUMPER-7-DLL_INFO :
/pkg/lib/lib_conaux_error.dll  0xfc1ee114  0x00000e78  0xfc295f64  0x00000088        0
RP/0/RP0/CPU0Sep 22 01:40:27.007 : dumper[54]: %DUMPER-7-DLL_INFO :
/pkg/lib/libttyutil.dll  0xfc2c7000  0x00003078  0xfc2baa48  0x00000168        0
RP/0/RP0/CPU0Sep 22 01:40:27.007 : dumper[54]: %DUMPER-7-DLL_INFO : /pkg/lib/libbag.dll
0xfc431000  0x0000ee98  0xfc40cc94  0x00000368        0
RP/0/RP0/CPU0Sep 22 01:40:27.008 : dumper[54]: %DUMPER-7-DLL_INFO : /pkg/lib/libchkpt.dll
0xfc474000  0x0002ecf8  0xfc4a3000  0x00000950        0
RP/0/RP0/CPU0Sep 22 01:40:27.008 : dumper[54]: %DUMPER-7-DLL_INFO :
/pkg/lib/libsysdbbackend.dll  0xfc8ed000  0x0000997c  0xfc8d3aa8  0x0000028c        0
RP/0/RP0/CPU0Sep 22 01:40:27.008 : dumper[54]: %DUMPER-7-DLL_INFO :
/pkg/lib/libttymgmtconnection.dll  0xfce85000  0x00004208  0xfce8a000  0x00000468        0
RP/0/RP0/CPU0Sep 22 01:40:27.009 : dumper[54]: %DUMPER-7-DLL_INFO :
/pkg/lib/libttymgmt.dll  0xfcea4000  0x0000e944  0xfce8abf0  0x000003c8        0
RP/0/RP0/CPU0Sep 22 01:40:27.009 : dumper[54]: %DUMPER-7-DLL_INFO :
/pkg/lib/libttynmspc.dll  0xfcec7000  0x00004a70  0xfcec6644  0x000002c8        0
RP/0/RP0/CPU0Sep 22 01:40:28.396 : dumper[54]: %DUMPER-5-CORE_FILE_NAME : Core for process
pkg/bin/devc-conaux at
harddisk:/coredump/devc-conaux.by.dumper_gen.sparse.20040922-014027.node0_RP0_CPU0.ppc.Z
RP/0/RP0/CPU0Sep 22 01:40:32.309 : dumper[54]: %DUMPER-5-DUMP_SUCCESS : Core dump success
```

exception filepath

To modify core dump settings, use the **exception filepath** command in administration configuration mode or in global configuration mode. To remove the configuration, use the **no** form of this command.

exception [**choice** *preference*] [**compress** {**on** | **off**}] [**filename** *filename lower-limit*-*higher-limit*] **filepath** *filepath*

no exception [**choice** *preference*] [**compress** {**on** | **off**}] [**filename** *filename lower-limit*-*higher-limit*] [**filepath** *filepath*]

Syntax Description

choice *preference*	(Optional) Configures the order of preference for the destination of core dump files. Up to the three destinations can be defined. Valid values are 1 to 3.
compress {**on** | **off**}	(Optional) Specifies whether or not the core dump file should be sent compressed. By default, core dump files are sent compressed. If you specify the **compress** keyword, you must specify one of the following required keywords: • **on**—Compresses the core dump file before sending it. • **off**—Does not compress the core dump file before sending it.
filename *filename lower-limit*-*higher-limit*	(Optional) Specifies the filename to be appended to core dump files and the lower and higher limit range of core dump files to be sent to a specified destination before being recycled by the circular buffer. See Table 65 in the "Usage Guidelines" section for a description of the default core dump file naming convention. Valid values for the *lower-limit* argument are 0 to 4. Valid values for the *higher-limit* argument are 5 to 64. A hyphen (**-**) must immediately follow the *lower-limit* argument. **Note** To uniquely identify each core dump file, a value is appended to each core dump file, beginning with the lower limit value configured for the *lower-limit* argument and continuing until the higher limit value configured for the *higher-limit* argument has been reached. After the higher limit value has been reached, the Cisco IOS XR software begins to recycle the values appended to core dump files, beginning with the lower limit value.
filepath	Local file system or network protocol, followed by the directory path. All local file systems are supported. The following network protocols are supported: TFTP and FTP.

Defaults

If you do not specify the order of preference for the destination of core dump files using the **choice** keyword and *preference* argument, the default preference is the primary location (that is, **choice 1**). Core dump files are sent compressed.

The default filenaming convention used for core dump files is described in the "Usage Guidelines" section. See Table 65.

Command Modes

Administration configuration
Global configuration

Command History

Release	Modification
Release 2.0	This command was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	This command was first supported on the Cisco XR 12000 Series Router.
Release 3.3.0	No modification.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

Use the **exception filepath** command to modify core dump settings, such as the destination file path to store core dump files, file compression, and the filename appended to core dumps.

Up to three user-defined locations may be configured as the preferred destinations for core dump files:

- Primary location—The primary destination for core dump files. Enter the **choice** keyword and a value of **1** (that is, **choice 1**) for the *preference* argument to specify a destination as the primary location for core dump files.
- Secondary location—The secondary fallback choice for the destination for core dump files, if the primary location is unavailable (for example, if the hard disk is set as the primary location and the hard disk fails). Enter the **choice** keyword and a value of **2** (that is, **choice 2**) for the *preference* argument to specify a destination as the secondary location for core dump files.
- Tertiary location—The tertiary fallback choice as the destination for core dump files, if the primary and secondary locations fail. Enter the **choice** keyword and a value of 3 (that is, **choice 3**) for the *preference* argument to specify a destination as the tertiary location for core dump files.

When specifying a destination for a core dump file, you can specify an absolute file path on a local file system or on a network server. The following network protocols are supported: TFTP and FTP.

Note We recommend that you specify a location on the hard disk as the primary location on the Cisco CRS-1 router.

In addition to the three preferred destinations that can be configured, Cisco IOS XR software provides three default fallback destinations for core dump files in the event that user-defined locations are unavailable.

The default fallback destinations on the Cisco CRS-1 are:

- harddisk:/dumper
- disk1:/dumper
- disk0:/dumper

The default fallback destinations on Cisco XR 12000 Series Routers are:

- disk1:/dumper
- disk0:/dumper
- bootflash:/dumper

Note If a default destination is a boot device, the core dump file is not sent to that destination.

We recommend that you configure at least one preferred destination for core dump files as a preventive measure if the default fallback paths are unavailable. Configuring at least one preferred destination also ensures that core dump files are archived because the default fallback destinations store only the first and last core dump files for a crashed process.

Note The Cisco IOS XR software does not save a core file on a local storage device if the size of the core dump file creates a low-memory condition.

By default, the Cisco IOS XR software assigns filenames to core dump files according to the following format:

process[.by.*requester* |.abort][.sparse].*date-time.node.processor-type*[.Z]

For example:

```
packet.by.dumper_gen.20040921-024800.node0_RP0_CPU0.ppc.Z
```

Table 65 describes the default core dump filenaming convention.

Table 65 Default Core Dump Filenaming Convention Description

Field	Description
process	Name of the process that generated the core dump.
.by.*requester* | .abort	If the core dump was generated because of a request by a process (requester), the core filename contains the string ".by.*requester*" where the *requester* variable is the name or process ID (pid) of the process that requested the core dump. If the core dump was due to a self-generated abort call request, the core filename contains the string ".abort" instead of the name of the requester.
.sparse	If a sparse core dump was generated instead of a full core dump, ".sparse" appears in the core dump filename.
.date-time	Date and time the dumper process was called by the process manager to generate the core dump. The *.date-time* time-stamp variable is expressed in the yyyy.mm.dd-hh.mm.ss format. Including the time stamp in the filename uniquely identifies the core dump filename.
.node	Node ID, expressed in the *rack/slot/module* notation, where the process that generated the core dump was running.
.processor-type	Type of processor (mips or ppc).
.Z	If the core dump was sent compressed, the filename contains the .Z suffix.

You can modify the default naming convention by specifying a filename to be appended to core dump files with the optional **filename** keyword and *filename* argument and by specifying a lower and higher limit ranges of values to be appended to core dump filenames with the *lower-limit* and *higher-limit* arguments, respectively. The filename that you specify for the *filename* argument is appended to the core dump file and the lower and higher limit ranges of core dump files to be sent to a specified destination before the filenames are recycled. Valid values for the *lower-limit* argument are 0 to 4. Valid values for the *higher-limit* argument are 5 to 64. A hyphen (**-**) must immediately follow the *lower-limit* argument. In addition, to uniquely identify each core dump file, a value is appended to each core dump file, beginning with the lower-limit value specified with the *lower-limit* argument and continuing until the higher-limit value specified with the *higher-limit* keyword has been reached. When the configured higher-limit value has been reached, the Cisco IOS XR software begins to recycle the values appended to core dump files, beginning with the lower-limit value.

Task ID

Task ID	Operations
diag	read, write

Examples

The following example shows how to configure the core dump setting for the primary user-defined preferred location. In this example, core files are configured to be sent uncompressed; the filename of core dump files is set to “core” (that is, all core filenames will be named core); the range value is set from 0 to 5 (that is, the values 0 to 5 are appended to the filename for the first five generated core dump files, respectively, before being recycled); and the destination is set to a directory on the hard disk.

```
RP/0/RP0/CPU0:router(config)# exception choice 1 compress off filename core 0-5 filepath
/harddisk:/corefile
```

Related Commands

Command	Description
exception pakmem	Collects packet memory information in core dumps.
exception sparse	Enables or disables sparse core dumps.
exception sprsize	Sets the maximum size of core dump files.
show exception	Displays the configured core dump settings.

exception pakmem

To configure the collection of packet memory information in core dump files, use the **exception pakmem** command in administration configuration mode or in global configuration mode. To remove the configuration, use the **no** form of this command.

exception pakmem {**on** | **off**}

no exception pakmem {**on** | **off**}

Syntax Description

on	Enables the collection of packet memory information in core dump files.
off	Disables the collection of packet memory information in core dump files.

Defaults

Packet memory information is not included in core dump files.

Command Modes

Administration configuration
Global configuration

Command History

Release	Modification
Release 2.0	This command was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	This command was first supported on the Cisco XR 12000 Series Router.
Release 3.3.0	No modification.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

Use the **exception pakmem on** command to configure the collection of packet memory information in core dump files. Cisco Technical Support Center engineers and development engineers use packet memory information to debug packet memory issues related to a process.

Caution Including packet memory information in core dump files significantly increases the amount of data generated in the core dump file, which may delay the restart time for the process.

Task ID

Task ID	Operations
diag	read, write

Examples

The following example shows how to configure core dumps to include packet memory information:

```
RP/0/RP0/CPU0:router(config)# exception pakmem on
```

Related Commands

Command	Description
exception filepath	Modifies core dump settings.
exception sparse	Enables or disables sparse core dumps.
exception sprsize	Sets the maximum size of core dump files.
show exception	Displays the configured core dump settings.

exception sparse

To enable or disable sparse core dumps, use the **exception sparse** command in administration configuration mode or in global configuration mode. To remove the configuration, use the **no** form of this command.

exception sparse {**on** | **off**}

no exception sparse

Syntax Description

on	Enables sparse core dumps.
off	Disables sparse core dumps

Defaults

Sparse core dumps are disabled.

Command Modes

Administration configuration
Global configuration

Command History

Release	Modification
Release 2.0	This command was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	This command was first supported on the Cisco XR 12000 Series Router.
Release 3.3.0	No modification.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

Use the **exception sparse** command to reduce the amount of data generated in the core dump file. Sparse core dumps, thus, reduce the amount of time required to generate the core dump file because only referenced data is generated in the core file (at the cost of lost information in the core file). Reducing the time required to generate core dump files corresponds to faster process restart times.

Sparse core dumps contain the following information about crashed processes:

- Register information for all threads, and any memory pages referenced in these register values
- Stack information for all threads, and any memory pages referenced in these threads
- All memory pages referenced by a loaded dynamic loadable library (DLL) data section, if the final program counter falls in a DLL data section

- Any user-specified marker pages from the lib_dumper_marker DLL

The **exception sparse** command dumps memory pages based on trigger addresses found in the previously listed dump information, according to the following criteria:

- If the trigger address in the memory page is in the beginning 128 bytes of the memory page, the previous memory page in the continuous address region is dumped also.
- If the trigger address in the memory page is in the final 128 bytes of the memory page, the next memory page in the continuous address region is dumped also.
- In all other instances, only the memory page that includes the trigger address is dumped.

Task ID

Task ID	Operations
diag	read, write

Examples

The following example shows how to enable sparse core dumps:

```
RP/0/RP0/CPU0:router(config)# exception sparse on
```

Related Commands

Command	Description
exception filepath	Modifies core dump settings.
exception pakmem	Collects packet memory information in core dumps.
exception sprsize	Sets the maximum file size of core dump files.
show exception	Displays the configured core dump settings.

exception sprsize

To specify the maximum file size for core dumps, use the **exception sprsize** command in administration configuration mode or in global configuration mode. To remove the configuration, use the **no** form of this command.

exception sprsize *megabytes*

no exception sprsize

Syntax Description

megabytes	Size in megabytes (MB).

Defaults

megabytes: 192 MB

Command Modes

Administration configuration
Global configuration

Command History

Release	Modification
Release 2.0	This command was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	This command was first supported on the Cisco XR 12000 Series Router.
Release 3.3.0	No modification.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

Use the **exception sprsize** command to specify the maximum file size for core dumps. The maximum file size configured for the *megabytes* argument is used with the configuration set for the **exception sparse** command to determine whether or not to generate a sparse core dump file. If sparse core dumps are disabled and a core dump file is predicted to exceed the default value (192 MB) uncompressed or the value specified for the *megabytes* argument uncompressed, a sparse core dump file is generated. If sparse core dumps are enabled, a sparse core dump file is generated, regardless of the size of the core dump file.

Task ID

Task ID	Operations
diag	read, write

Examples

The following example shows how to set the file size of sparse core dumps to 300 MB:

```
RP/0/RP0/CPU0:router(config)# exception sprsize 300
```

Related Commands

Command	Description
exception sparse	Enables or disables sparse core dumps.

follow

To unobtrusively debug a live process or a live thread in a process, use the **follow** command in EXEC mode.

follow {**job** *job-id* | **process** *pid* | **location** *node-id*} [**all**] [**blocked**] [**debug** *level*] [**delay** *seconds*] [**dump** *address size*] [**iteration** *count*] [**priority** *level*] [**stackonly**] [**thread** *tid*] [**verbose**]

Syntax Description

job *job-id*	Follows a process by job ID.
process *pid*	Follows the process with the process ID (PID) specified for the *pid* argument.
location *node-id*	(Optional) Follows the target process on the designated node. The *node-id* argument is expressed in the *rack*/*slot*/*module* notation.
all	(Optional) Follows all threads.
blocked	(Optional) Follows the chain of thread IDs (TIDs) or PIDs that are blocking the target process.
debug *level*	(Optional) Sets the debug level for the following operation. Valid values for the level argument are 0 to 10.
delay *seconds*	(Optional) Sets the delay interval between each iteration. Valid values for the *seconds* argument are 0 to 255 seconds.
dump *address size*	(Optional) Dumps the memory segment starting with the specified memory address and size specified for the *address* and *size* arguments.
iteration *count*	(Optional) Specifies the number of times to display information. Valid values for the *count* argument are 0 to 255 iterations.
priority *level*	(Optional) Sets the priority level for the following operation. Valid values for the *level* argument are 1 to 63.
stackonly	(Optional) Displays only stack trace information.
thread *tid*	(Optional) Follows the TID of a process or job ID specified for the *tid* argument.
verbose	(Optional) Displays register and status information pertaining to the target process.

Defaults

Entering the **follow** command without any optional keywords or arguments performs the operation for five iterations from the local node with a delay of 5 seconds between each iteration. The output includes information about all live threads. This command uses the default scheduling priority from where the command is being run.

Command Modes

EXEC

Command History

Release	Modification
Release 3.2	This command was introduced on the Cisco CRS-1 and the Cisco XR 12000 Series Router.
Release 3.3.0	No modification.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

Use this command to unintrusively debug a live process or a live thread in a process. This command is particularly useful for debugging deadlock and livelock conditions, for examining the contents of a memory location or a variable in a process to determine the cause of a corruption issue, or in investigating issues where a thread is stuck spinning in a loop. A livelock condition is one that occurs when two or more processes continually change their state in response to changes in the other processes.

The following actions can be specified with this command:

- Follow all live threads of a given process or a given thread of a process and print stack trace in a format similar to core dump output.
- Follow a process in a loop for a given number of iterations.
- Set a delay between two iterations while invoking the command.
- Set the priority at which this process should run while this command is being run.
- Dump memory from a given virtual memory location for a given size.
- Display register values and status information of the target process.

Take a snapshot of the execution path of a thread asynchronously to investigate performance-related issues by specifying a high number of iterations with a zero delay.

Task ID

Task ID	Operations
basic-services	read

Examples

The following example shows how to use the **follow** command to debug the process associated with job ID 257 for one iteration:

```
RP/0/RP0/CPU0:router# follow job 257 iteration 1

Attaching to process pid = 28703 (pkg/bin/packet)
No tid specified, following all threads

DLL Loaded by this process
------------------------------

DLL path                Text addr. Text size  Data addr. Data size  Version
/pkg/lib/libovl.dll     0xfc0c9000 0x0000c398 0xfc0c31f0 0x0000076c        0
```

```
/pkg/lib/libplatform.dll 0xfc0d6000 0x0000aa88 0xfc0e1000 0x00002000        0
/pkg/lib/libsysmgr.dll   0xfc0e3000 0x0000aeac 0xfc0c395c 0x00000388        0
/pkg/lib/libinfra.dll    0xfc0ee000 0x000332ec 0xfc122000 0x00000c70        0
/pkg/lib/libios.dll      0xfc123000 0x0002c4bc 0xfc150000 0x00002000        0
/pkg/lib/libc.dll        0xfc152000 0x00077ae0 0xfc1ca000 0x00002000        0
/pkg/lib/libsyslog.dll   0xfc1d4000 0x0000530c 0xfc122c70 0x00000308        0
/pkg/lib/libbackplane.dll 0xfc1da000 0x0000134c 0xfc0c3e6c 0x000000a8        0
/pkg/lib/libnodeid.dll   0xfc1e7000 0x000091fc 0xfc1e61a8 0x00000208        0
/pkg/lib/libdebug.dll    0xfc23e000 0x0000ef64 0xfc1e6680 0x00000550        0
/pkg/lib/lib_procfs_util.dll 0xfc24d000 0x00004e2c 0xfc1e6bd0 0x000002a8        0
/pkg/lib/libsysdb.dll    0xfc252000 0x00046224 0xfc299000 0x0000079c        0
/pkg/lib/libsysdbutils.dll 0xfc29a000 0x0000ae04 0xfc29979c 0x000003ec        0
/pkg/lib/libwd_evm.dll   0xfc2a9000 0x0000481c 0xfc299b88 0x00000188        0
/pkg/lib/lib_mutex_monitor.dll 0xfc35e000 0x00002414 0xfc340850 0x00000128        0
/pkg/lib/libchkpt.dll    0xfc477000 0x0002ee04 0xfc474388 0x00000950        0
/pkg/lib/libpacket_common.dll 0xfc617000 0x000130f0 0xfc6056a0 0x000007b0        0

Iteration 1 of 1
------------------------------

Current process = "pkg/bin/packet", PID = 28703 TID = 1

trace_back: #0 0xfc1106dc [MsgReceivev]
trace_back: #1 0xfc0fc840 [msg_receivev]
trace_back: #2 0xfc0fc64c [msg_receive]
trace_back: #3 0xfc0ffa70 [event_dispatch]
trace_back: #4 0xfc0ffc2c [event_block]
trace_back: #5 0x48204410 [<N/A>]

ENDOFSTACKTRACE

Current process = "pkg/bin/packet", PID = 28703 TID = 2

trace_back: #0 0xfc1106dc [MsgReceivev]
trace_back: #1 0xfc0fc840 [msg_receivev]
trace_back: #2 0xfc0fc64c [msg_receive]
trace_back: #3 0xfc0ffa70 [event_dispatch]
trace_back: #4 0xfc0ffc2c [event_block]
trace_back: #5 0xfc48d848 [chk_evm_thread]

ENDOFSTACKTRACE

Current process = "pkg/bin/packet", PID = 28703 TID = 3

trace_back: #0 0xfc17d54c [SignalWaitinfo]
trace_back: #1 0xfc161c64 [sigwaitinfo]
trace_back: #2 0xfc10302c [event_signal_thread]

ENDOFSTACKTRACE

Current process = "pkg/bin/packet", PID = 28703 TID = 4

trace_back: #0 0xfc1106c4 [MsgReceivePulse]
trace_back: #1 0xfc0fc604 [msg_receive_async]
trace_back: #2 0xfc0ffa70 [event_dispatch]
trace_back: #3 0xfc0ffc5c [event_block_async]
trace_back: #4 0xfc35e36c [receive_events]

ENDOFSTACKTRACE

Current process = "pkg/bin/packet", PID = 28703 TID = 5

trace_back: #0 0xfc17d564 [SignalWaitinfo_r]
trace_back: #1 0xfc161c28 [sigwait]
```

```
trace_back: #2 0x48203928 [<N/A>]

ENDOFSTACKTRACE
```

The following example shows how to use the **follow** command to debug TID 5 of the process associated with job ID 257 for one iteration:

```
RP/0/RP0/CPU0:router# follow job 257 iteration 1 thread 5

Attaching to process pid = 28703 (pkg/bin/packet)

DLL Loaded by this process
------------------------------

DLL path                 Text addr. Text size  Data addr. Data size  Version
/pkg/lib/libovl.dll      0xfc0c9000 0x0000c398 0xfc0c31f0 0x0000076c        0
/pkg/lib/libplatform.dll 0xfc0d6000 0x0000aa88 0xfc0e1000 0x00002000        0
/pkg/lib/libsysmgr.dll   0xfc0e3000 0x0000aeac 0xfc0c395c 0x00000388        0
/pkg/lib/libinfra.dll    0xfc0ee000 0x000332ec 0xfc122000 0x00000c70        0
/pkg/lib/libios.dll      0xfc123000 0x0002c4bc 0xfc150000 0x00002000        0
/pkg/lib/libc.dll        0xfc152000 0x00077ae0 0xfc1ca000 0x00002000        0
/pkg/lib/libsyslog.dll   0xfc1d4000 0x0000530c 0xfc122c70 0x00000308        0
/pkg/lib/libbackplane.dll 0xfc1da000 0x0000134c 0xfc0c3e6c 0x000000a8        0
/pkg/lib/libnodeid.dll   0xfc1e7000 0x000091fc 0xfc1e61a8 0x00000208        0
/pkg/lib/libdebug.dll    0xfc23e000 0x0000ef64 0xfc1e6680 0x00000550        0
/pkg/lib/lib_procfs_util.dll 0xfc24d000 0x00004e2c 0xfc1e6bd0 0x000002a8        0
/pkg/lib/libsysdb.dll    0xfc252000 0x00046224 0xfc299000 0x0000079c        0
/pkg/lib/libsysdbutils.dll 0xfc29a000 0x0000ae04 0xfc29979c 0x000003ec        0
/pkg/lib/libwd_evm.dll   0xfc2a9000 0x0000481c 0xfc299b88 0x00000188        0
/pkg/lib/lib_mutex_monitor.dll 0xfc35e000 0x00002414 0xfc340850 0x00000128        0
/pkg/lib/libchkpt.dll    0xfc477000 0x0002ee04 0xfc474388 0x00000950        0
/pkg/lib/libpacket_common.dll 0xfc617000 0x000130f0 0xfc6056a0 0x000007b0        0

Iteration 1 of 1
------------------------------

Current process = "pkg/bin/packet", PID = 28703 TID = 5

trace_back: #0 0xfc17d564 [SignalWaitinfo_r]
trace_back: #1 0xfc161c28 [sigwait]
trace_back: #2 0x48203928 [<N/A>]

ENDOFSTACKTRACE
```

The following example shows how to use the **follow** command to debug the chain of threads blocking thread 2 associated with the process assigned PID 139406:

```
RP/0/RP0/CPU0:router# follow process 139406 blocked iteration 1 thread 2

Attaching to process pid = 139406 (pkg/bin/lpts_fm)

DLL Loaded by this process
------------------------------

DLL path                 Text addr. Text size  Data addr. Data size  Version
/pkg/lib/libplatform.dll 0xfc0d6000 0x0000aa88 0xfc0e1000 0x00002000        0
/pkg/lib/libsysmgr.dll   0xfc0e3000 0x0000aeac 0xfc0c395c 0x00000388        0
/pkg/lib/libinfra.dll    0xfc0ee000 0x000332ec 0xfc122000 0x00000c70        0
/pkg/lib/libios.dll      0xfc123000 0x0002c4bc 0xfc150000 0x00002000        0
/pkg/lib/libc.dll        0xfc152000 0x00077ae0 0xfc1ca000 0x00002000        0
/pkg/lib/libltrace.dll   0xfc1cc000 0x00007f5c 0xfc0c3ce4 0x00000188        0
/pkg/lib/libsyslog.dll   0xfc1d4000 0x0000530c 0xfc122c70 0x00000308        0
/pkg/lib/libbackplane.dll 0xfc1da000 0x0000134c 0xfc0c3e6c 0x000000a8        0
/pkg/lib/libnodeid.dll   0xfc1e7000 0x000091fc 0xfc1e61a8 0x00000208        0
```

```
/pkg/lib/libdebug.dll    0xfc23e000 0x0000ef64 0xfc1e6680 0x00000550        0
/pkg/lib/lib_procfs_util.dll 0xfc24d000 0x00004e2c 0xfc1e6bd0 0x000002a8        0
/pkg/lib/libsysdb.dll    0xfc252000 0x00046224 0xfc299000 0x0000079c        0
/pkg/lib/libsysdbutils.dll 0xfc29a000 0x0000ae04 0xfc29979c 0x000003ec        0
/pkg/lib/libwd_evm.dll   0xfc2a9000 0x0000481c 0xfc299b88 0x00000188        0
/pkg/lib/libbag.dll      0xfc40c000 0x0000ee98 0xfc41b000 0x00000368        0
/pkg/lib/libwd_notif.dll 0xfc4f8000 0x00005000 0xfc4fd000 0x00001000        0
/pkg/lib/libifmgr.dll    0xfc665000 0x00029780 0xfc68f000 0x00003000        0
/pkg/lib/libnetio_client.dll 0xfca6a000 0x000065c8 0xfca2c4f8 0x000001b4        0
/pkg/lib/libpa_client.dll 0xfcec5000 0x00006e9c 0xfcecc000 0x00003000        0
/pkg/lib/libltimes.dll   0xfcecf000 0x00002964 0xfcdc4f20 0x000000a8        0

Iteration 1 of 1
-------------------------------

Current process = "pkg/bin/lpts_fm", PID = 139406 TID = 2

trace_back: #0 0xfc110744 [MsgSendv]
trace_back: #1 0xfc0fbf04 [msg_sendv]
trace_back: #2 0xfc0fbbd8 [msg_send]
trace_back: #3 0xfcec7580 [pa_fm_close]
trace_back: #4 0xfcec78b0 [pa_fm_process_0]

ENDOFSTACKTRACE

REPLY (node node0_RP1_CPU0, pid 57433)

No specific TID, following all threads of 57433 (pkg/bin/lpts_pa)
---------------------------------------------------------------

DLL Loaded by this process
-------------------------------

DLL path                 Text addr. Text size  Data addr. Data size  Version
/pkg/lib/libplatform.dll 0xfc0d6000 0x0000aa88 0xfc0e1000 0x00002000         0
/pkg/lib/libsysmgr.dll   0xfc0e3000 0x0000aeac 0xfc0c395c 0x00000388         0
/pkg/lib/libinfra.dll    0xfc0ee000 0x000332ec 0xfc122000 0x00000c70         0
/pkg/lib/libios.dll      0xfc123000 0x0002c4bc 0xfc150000 0x00002000         0
/pkg/lib/libc.dll        0xfc152000 0x00077ae0 0xfc1ca000 0x00002000         0
/pkg/lib/libltrace.dll   0xfc1cc000 0x00007f5c 0xfc0c3ce4 0x00000188         0
/pkg/lib/libsyslog.dll   0xfc1d4000 0x0000530c 0xfc122c70 0x00000308         0
/pkg/lib/libbackplane.dll 0xfc1da000 0x0000134c 0xfc0c3e6c 0x000000a8        0
/pkg/lib/libnodeid.dll   0xfc1e7000 0x000091fc 0xfc1e61a8 0x00000208         0
/pkg/lib/libdebug.dll    0xfc23e000 0x0000ef64 0xfc1e6680 0x00000550         0
/pkg/lib/lib_procfs_util.dll 0xfc24d000 0x00004e2c 0xfc1e6bd0 0x000002a8        0
/pkg/lib/libsysdb.dll    0xfc252000 0x00046224 0xfc299000 0x0000079c         0
/pkg/lib/libsysdbutils.dll 0xfc29a000 0x0000ae04 0xfc29979c 0x000003ec        0
/pkg/lib/libwd_evm.dll   0xfc2a9000 0x0000481c 0xfc299b88 0x00000188         0
/pkg/lib/lrdlib.dll      0xfc2f6000 0x0000a900 0xfc2f551c 0x00000610         0
/pkg/lib/liblrfuncs.dll  0xfc30e000 0x00001998 0xfc2ebd80 0x000001ec         0
/pkg/lib/libdscapi.dll   0xfc310000 0x0000457c 0xfc2f5b2c 0x0000035c         0
/pkg/lib/liblrdshared.dll 0xfc315000 0x00005fec 0xfc31b000 0x00002000        0
/pkg/lib/libbag.dll      0xfc40c000 0x0000ee98 0xfc41b000 0x00000368         0
/pkg/lib/libchkpt.dll    0xfc477000 0x0002ee04 0xfc474388 0x00000950         0
/pkg/lib/libwd_notif.dll 0xfc4f8000 0x00005000 0xfc4fd000 0x00001000         0
/pkg/lib/libltrace_sdt.dll 0xfc65c000 0x000034fc 0xfc65b73c 0x00000568        0
/pkg/lib/libfabhandle.dll 0xfc6be000 0x00003354 0xfc65bca4 0x00000248        0
/pkg/lib/libfsdb_ltrace_util_rt.dll 0xfc6ea000 0x00001b74 0xfc605e50 0x00000108        0
/pkg/lib/libbcdl.dll     0xfc6fb000 0x0000f220 0xfc6fa6e8 0x0000045c         0
/pkg/lib/liblpts_pa_fgid.dll 0xfc8d7000 0x00006640 0xfc7acd5c 0x00000208        0
/pkg/lib/libfgid.dll     0xfc910000 0x0001529c 0xfc926000 0x00002000         0
/pkg/lib/libltimes.dll   0xfcecf000 0x00002964 0xfcdc4f20 0x000000a8         0

Current process = "pkg/bin/lpts_pa", PID = 57433 TID = 1
```

```
trace_back: #0 0xfc1106dc [MsgReceivev]
trace_back: #1 0xfc0fc840 [msg_receivev]
trace_back: #2 0xfc0fc64c [msg_receive]
trace_back: #3 0xfc0ffa70 [event_dispatch]
trace_back: #4 0xfc0ffc2c [event_block]
trace_back: #5 0x48201904 [<N/A>]
trace_back: #6 0x48201e3c [<N/A>]

ENDOFSTACKTRACE

Current process = "pkg/bin/lpts_pa", PID = 57433 TID = 2

trace_back: #0 0xfc1106dc [MsgReceivev]
trace_back: #1 0xfc0fc840 [msg_receivev]
trace_back: #2 0xfc0fc64c [msg_receive]
trace_back: #3 0xfc0ffa70 [event_dispatch]
trace_back: #4 0xfc0ffc2c [event_block]
trace_back: #5 0x4821e978 [<N/A>]

ENDOFSTACKTRACE

Current process = "pkg/bin/lpts_pa", PID = 57433 TID = 3

trace_back: #0 0xfc1106dc [MsgReceivev]
trace_back: #1 0xfc0fc840 [msg_receivev]
trace_back: #2 0xfc0fc64c [msg_receive]
trace_back: #3 0xfc0ffa70 [event_dispatch]
trace_back: #4 0xfc0ffc2c [event_block]
trace_back: #5 0x482064c4 [<N/A>]

ENDOFSTACKTRACE
```

The following example shows how to use the **follow** command to debug the chain of threads blocking thread 2 associated with the process assigned PID 139406:

```
RP/0/RP0/CPU0:router# follow process 139406 blocked iteration 1 stackonly thread 2

Attaching to process pid = 139406 (pkg/bin/lpts_fm)

Iteration 1 of 1
------------------------------

Current process = "pkg/bin/lpts_fm", PID = 139406 TID = 2

trace_back: #0 0xfc110744 [MsgSendv]
trace_back: #1 0xfc0fbf04 [msg_sendv]
trace_back: #2 0xfc0fbbd8 [msg_send]
trace_back: #3 0xfcec7580 [pa_fm_close]
trace_back: #4 0xfcec78b0 [pa_fm_process_0]

ENDOFSTACKTRACE

REPLY (node node0_RP1_CPU0, pid 57433)

No specific TID, following all threads of 57433 (pkg/bin/lpts_pa)
---------------------------------------------------------------------

Current process = "pkg/bin/lpts_pa", PID = 57433 TID = 1

trace_back: #0 0xfc1106dc [MsgReceivev]
trace_back: #1 0xfc0fc840 [msg_receivev]
trace_back: #2 0xfc0fc64c [msg_receive]
trace_back: #3 0xfc0ffa70 [event_dispatch]
```

```
trace_back: #4 0xfc0ffc2c [event_block]
trace_back: #5 0x48201904 [<N/A>]
trace_back: #6 0x48201e3c [<N/A>]

ENDOFSTACKTRACE

Current process = "pkg/bin/lpts_pa", PID = 57433 TID = 2

trace_back: #0 0xfc1106dc [MsgReceivev]
trace_back: #1 0xfc0fc840 [msg_receivev]
trace_back: #2 0xfc0fc64c [msg_receive]
trace_back: #3 0xfc0ffa70 [event_dispatch]
trace_back: #4 0xfc0ffc2c [event_block]
trace_back: #5 0x4821e978 [<N/A>]

ENDOFSTACKTRACE

Current process = "pkg/bin/lpts_pa", PID = 57433 TID = 3

trace_back: #0 0xfc1106dc [MsgReceivev]
trace_back: #1 0xfc0fc840 [msg_receivev]
trace_back: #2 0xfc0fc64c [msg_receive]
trace_back: #3 0xfc0ffa70 [event_dispatch]
trace_back: #4 0xfc0ffc2c [event_block]
trace_back: #5 0x482064c4 [<N/A>]

ENDOFSTACKTRACE
```

Related Commands

Command	Description
show processes	Displays information about the running processes.

monitor processes

To display auto-updating statistics on processes in a full-screen mode, use the **monitor processes** command in administration EXEC mode or in EXEC mode.

monitor processes [**dumbtty**] [**location** *node-id*]

Syntax Description

dumbtty	(Optional) Displays the output of the command as if on a dumb terminal (the screen is not refreshed).
location *node-id*	(Optional) Displays the output of the command from the designated node. The *node-id* argument is entered in the *rack*/*slot*/*module* notation.

Defaults

If you omit all keywords, the command displays the top 10 processes of CPU usage for the local node, sorted in descending order by the time used. The display is cleared and updated every 5 seconds until you quit the **monitor processes** command by pressing the **q** key.

Command Modes

Administration EXEC
EXEC

Command History

Release	Modification
Release 2.0	This command was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	This command was first supported on the Cisco XR 12000 Series Router. The command was made available in administration EXEC mode.
Release 3.3.0	No modification.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

Use the **monitor processes** command to display the top ten processes based on CPU usage. The display refreshes every 10 seconds.

- To change the parameters displayed by the **monitor processes** command, enter one of the interactive commands described in Table 66.
- To terminate the display and return to the system prompt, enter the **q** key.
- To list the interactive commands, type **?** during the display.

The available interactive commands are described in Table 66.

Table 66 Interactive Commands Available for the monitor processes Command

Command	Description
?	Displays the available interactive commands.
c	Sorts display by number of open channels.
d	Changes the delay interval between updates.
f	Sorts display by number of open files.
k	Kills a process.
l	Refreshes the screen.
m	Sorts display by memory used.
n	Changes the number of processes to be displayed.
q	Quits the interactive display and returns the prompt to EXEC mode.
t	Sorts display by time (default).

Task ID

Task ID	Operations
basic-services	execute

Examples

The following is sample output from the **monitor processes** command:

```
RP/0/RP0/CPU0:router# monitor processes

195 processes; 628 threads; 3300 channels, 4579 fds
CPU states: 47.6% idle, 1.2% user, 51.1% kernel
Memory: 2048M total, 1576M avail, page size 4K

      JID TIDS Chans   FDs Tmrs   MEM   HH:MM:SS   CPU  NAME
        1   27  198     8    1     0    5:53:31 51.11% kernel
       52    5  215    44    5   228K    0:00:02  0.52% devc-conaux
      342    4  195    14    6     1M    0:00:08  0.34% wdsysmon
   495806    1    1    10    0   648K    0:00:00  0.16% ptop
      293    7   31    39   11   352K    0:00:09  0.07% shelfmgr
       55   11   24    14    5    16M    0:00:29  0.06% eth_server
      121    3   10     8    2   564K    0:00:05  0.02% bcm_process
      311    4    7    18    4   216K    0:00:02  0.01% sysdb_medusa_s
      138    4   14    40    5   240K    0:00:01  0.01% devc-vty
      265    5   31    19    4   204K    0:00:09  0.01% packet
```

The following is sample output from the **monitor processes** command using the optional **location** keyword and *node-id* argument:

```
RP/0/RP0/CPU0:router# monitor processes location 0/RP0/CPU0

202 processes; 724 threads; 3750 channels, 5092 fds
CPU states: 48.8% idle, 0.8% user, 1.5% kernel
Memory: 2048M total, 1526M avail, page size 4K

      JID TIDS Chans   FDs Tmrs   MEM   HH:MM:SS   CPU  NAME
        1   27  205     3    1     0   10:54:12  1.52% procnto-600-smp-cisco-instr
      264    5   42    19    4   272K    0:00:15  0.37% packet
       53    2  202   564    0     1M    0:00:06  0.10% dllmgr
      180   15   93    42    6     1M    0:00:19  0.05% gsp
       69   22   94     8    3     1M    0:00:54  0.04% qnet
```

```
  67    5    4     6    0   956K    0:00:04  0.03% pkgfs
 156    2    6    18    1   480K    0:00:00  0.02% envmon
 294    1    6    12    1   112K    0:00:00  0.02% showd_lc
 314    3  185    14    4     1M    0:00:17  0.02% sysdb_svr_local
 310    4    7    18    4   276K    0:00:07  0.02% sysdb_medusa_s
```

The following is sample output from the **show processes** command using the **dumbtty** optional keyword:

```
RP/0/RP0/CPU0:router# monitor processes dumbtty

Computing times...195 processes; 628 threads; 3721 channels, 4801 fds
CPU states: 37.1% idle, 1.1% user, 61.7% kernel
Memory: 2048M total, 1576M avail, page size 4K

      JID TIDS Chans   FDs Tmrs   MEM   HH:MM:SS   CPU  NAME
        1   27  198     6    1     0    6:33:33 61.76% kernel
   544958    1    1     8    0   648K    0:00:00  0.64% ptop
      293    7   31    39   11   352K    0:00:10  0.10% shelfmgr
      180   15   82    42    6     5M    0:00:26  0.10% gsp
      304    3   14    29    7   304K    0:00:02  0.06% statsd_manager
       55   11   24    14    5    16M    0:00:32  0.03% eth_server
       70   22   91     8    3     1M    0:00:31  0.03% qnet
      153    2   35    18    4   120K    0:00:01  0.03% dsc
      303    3   25    34    5   292K    0:00:00  0.03% statsd_server
      121    3   10     8    2   564K    0:00:06  0.03% bcm_process

195 processes; 628 threads; 3409 channels, 4601 fds
CPU states: 46.5% idle, 0.5% user, 52.8% kernel
Memory: 2048M total, 1576M avail, page size 4K

      JID TIDS Chans   FDs Tmrs   MEM   HH:MM:SS   CPU  NAME
        1   27  198     6    1     0    6:33:44 52.89% kernel
       52    5  215    44    5   228K    0:00:06  0.38% devc-conaux
      309    6   25    23    8   352K    0:00:08  0.03% sysdb_mc
      315    3  177    14    4     1M    0:00:12  0.03% sysdb_svr_local
      138    4   14    40    5   240K    0:00:02  0.02% devc-vty
      298    9   25   111    9     2M    0:00:09  0.01% snmpd
       67    4    4     7    0   804K    0:00:04  0.00% pkgfs
       53    2  195   547    0   944K    0:00:06  0.00% dllmgr
      311    4    7    18    4   216K    0:00:03  0.00% sysdb_medusa_s
      342    4  195    14    6     1M    0:00:08  0.00% wdsysmon
```

Table 67 describes the significant fields shown in the display.

Table 67 monitor processes Field Descriptions

Field	Description
JID	Job ID.
TIDS	Thread ID.
Chans	Number of open channels.
FDs	Number of open file descriptors.
Tmrs	Number of timers.
MEM	Dynamic memory currently in use.
HH:MM:SS	Run time of process since last restart.
CPU	Percentage of CPU used by process thread.
NAME	Process name.

Using Interactive Commands

When the **n** or **d** interactive command is used, the **monitor processes** command prompts you to enter a number. For example, when the interactive command **n** is entered, the prompt responds as shown below:

```
Enter number of procs to display: 15

195 processes; 628 threads; 3375 channels, 4495 fds
CPU states: 49.0% idle, 0.9% user, 50.0% kernel
Memory: 2048M total, 1576M avail, page size 4K

      JID TIDS Chans   FDs Tmrs   MEM   HH:MM:SS   CPU  NAME
        1   27  198     2    1     0    6:11:43 50.01% kernel
       52    5  215    44    5  228K    0:00:05  0.72% devc-conaux
      293    7   31    39   11  352K    0:00:09  0.04% shelfmgr
      315    3  177    14    4    1M    0:00:11  0.03% sysdb_svr_local
      304    3   14    29    7  304K    0:00:01  0.02% statsd_manager
      309    6   25    23    8  352K    0:00:08  0.02% sysdb_mc
      342    4  195    14    6    1M    0:00:08  0.01% wdsysmon
      298    9   25   111    9    2M    0:00:09  0.00% snmpd
      265    5   31    19    4  204K    0:00:09  0.00% packet
      153    2   35    18    4  120K    0:00:00  0.00% dsc
      290    4    6    17    2  112K    0:00:00  0.00% sc_reddrv
      275    7   34    36    7  588K    0:00:00  0.00% qlink
      303    3   25    34    5  292K    0:00:00  0.00% statsd_server
      262    5   23    46    6    1M    0:00:00  0.00% ospf
      239    3   26    31    9  452K    0:00:00  0.00% lpts_pa
```

If the number you enter is outside the acceptable range, you are prompted for another number:

```
Enter number of procs to display: 435
Please enter a number between 5 and 40
Enter number of procs to display:
```

Related Commands

Command	Description
monitor threads	Displays auto-updating process and thread statistics in a full-screen mode.
show processes	Displays information about the running processes.

monitor threads

To display auto-updating statistics on threads in a full-screen mode, use the **monitor threads** command in administration EXEC mode or in EXEC mode.

monitor threads [**dumbtty**] [**iteration** *number*] [**location** *node-id*]

Syntax Description

dumbtty	(Optional) Displays the output of the command as if on a dumb terminal (the screen is not refreshed).
iteration *number*	(Optional) Number of times the statistics display is to be updated, in the range from 0 to 4294967295.
location *node-id*	(Optional) Displays the output from the command from the designated node. The *node-id* argument is entered in the *rack/slot/module* notation.

Defaults

When all keywords are omitted, the command displays the first ten threads for the local node, sorted in descending order by the time used. The display is cleared and updated every 5 seconds until you quit the **monitor threads** command.

Command Modes

Administration EXEC
EXEC

Command History

Release	Modification
Release 2.0	This command was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	This command was first supported on the Cisco XR 12000 Series Router. The command was made available in administration EXEC mode.
Release 3.3.0	No modification.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

Use the **monitor threads** command to show the top ten threads based on CPU usage. The display refreshes every 10 seconds.

- To change the parameters displayed by the **monitor threads** command, enter one of the key commands described in Table 68.
- To terminate the display and return to the system prompt, enter the **q** key.

- To list the interactive commands, type **?** during the display.

The available interactive display commands are described in Table 68.

Table 68 Interactive Display Commands for the monitor threads Command

Command	Description
?	Displays the available interactive commands.
d	Changes the delay interval between updates.
k	Kills a process.
l	Refreshes the screen.
n	Changes the number of threads to be displayed.
q	Quits the interactive display and returns the prompt to EXEC mode.

Task ID

Task ID	Operations
basic-services	execute

Examples

The following is sample output from the **monitor threads** command:

```
RP/0/RP0/CPU0:router# monitor threads

195 processes; 628 threads;
CPU states: 98.2% idle, 0.9% user, 0.7% kernel
Memory: 2048M total, 1576M avail, page size 4K

   JID   TID LAST_CPU PRI STATE  HH:MM:SS      CPU  COMMAND
     1    12   1       10 Rcv     0:00:09     0.42% procnto-600-smp-cisco-instr
     1    25   1       10 Run     0:00:30     0.36% procnto-600-smp-cisco-instr
   342     1   1       19 Rcv     0:00:07     0.20% wdsysmon
    52     5   0       21 Rcv     0:00:03     0.15% devc-conaux
    52     3   1       18 Rcv     0:00:02     0.07% devc-conaux
532670     1   0       10 Rply    0:00:00     0.07% top
   293     6   0       55 Rcv     0:00:06     0.03% shelfmgr
    55     8   0       10 Rcv     0:00:02     0.03% eth_server
   315     3   0       10 Rcv     0:00:11     0.03% sysdb_svr_local
    55     7   0       55 Rcv     0:00:11     0.02% eth_server
```

The following is sample output from the **monitor threads** command using the optional **location** keyword:

```
RP/0/RP0/CPU0:router# monitor threads location 0/RP0/CPU0

Computing times...195 processes; 628 threads;
CPU states: 95.1% idle, 2.7% user, 2.0% kernel
Memory: 2048M total, 1576M avail, page size 4K

   JID   TID LAST_CPU PRI STATE  HH:MM:SS      CPU  COMMAND
     1    25   0       10 Run     0:00:32     2.08% procnto-600-smp-cisco-instr
   265     5   0       10 SigW    0:00:09     0.89% packet
   279     1   1       10 Rcv     0:00:00     0.65% qsm
557246     1   0       10 Rply    0:00:00     0.51% top
   293     5   1       55 Rcv     0:00:01     0.07% shelfmgr
   180    13   1       10 Rcv     0:00:02     0.07% gsp
   315     3   0       10 Rcv     0:00:12     0.07% sysdb_svr_local
    55     7   1       55 Rcv     0:00:12     0.04% eth_server
```

```
   180      1    0       10 Rcv      0:00:01      0.04% gsp
   298      9    0       10 Rcv      0:00:01      0.04% snmpd
```

Table 69 describes the significant fields shown in the display.

Table 69 monitor threads Field Descriptions

Field	Description
JID	Job ID.
TIDS	Thread ID.
LAST_CPU	Number of open channels.
PRI	Priority level of the thread.
STATE	State of the thread.
HH:MM:SS	Run time of process since last restart.
CPU	Percentage of CPU used by process thread.
COMMAND	Process name.

Using Interactive Commands

When the **n** or **d** interactive command is used, the **monitor threads** command prompts for a number appropriate to the specific interactive command. The following is sample output from the **monitor threads** command showing the use of the interactive command **n** after the first display cycle to change the number of threads:

```
RP/0/RP0/CPU0:router# monitor threads

Computing times... 87 processes; 249 threads;
CPU states: 84.8% idle, 4.2% user, 10.9% kernel
Memory: 256M total, 175M avail, page size 4K

   JID   TID PRI STATE  HH:MM:SS      CPU  COMMAND
     1     6  10 Run      0:00:10    10.92% kernel
553049     1  10 Rply     0:00:00     4.20% top
    58     3  10 Rcv      0:00:24     0.00% sysdbsvr
     1     3  10 Rcv      0:00:21     0.00% kernel
    69     1  10 Rcv      0:00:20     0.00% wdsysmon
     1     5  10 Rcv      0:00:20     0.00% kernel
   159     2  10 Rcv      0:00:05     0.00% qnet
   160     1  10 Rcv      0:00:05     0.00% netio
   157     1  10 NSlp     0:00:04     0.00% envmon_periodic
   160     9  10 Intr     0:00:04     0.00% netio

n

Enter number of threads to display: 3
Please enter a number between 5 and 40
Enter number of threads to display: 8
87 processes; 249 threads;
CPU states: 95.3% idle, 2.9% user, 1.7% kernel
Memory: 256M total, 175M avail, page size 4K

   JID   TID PRI STATE  HH:MM:SS      CPU  COMMAND
     1     6  10 Run      0:00:11     1.76% kernel
    69     1  10 Rcv      0:00:20     1.11% wdsysmon
    58     3  10 Rcv      0:00:24     0.40% sysdbsvr
   157     1  10 NSlp     0:00:04     0.23% envmon_periodic
   159    19  10 Rcv      0:00:02     0.20% qnet
553049     1  10 Rply     0:00:00     0.20% top
```

```
   159    12  10 Rcv     0:00:03     0.13% qnet
   160     1  10 Rcv     0:00:05     0.10% netio
```

When a number outside the acceptable range is entered, the acceptable range is displayed:

```
Please enter a number between 5 and 40
Enter number of threads to display:
```

Related Commands

Command	Description
monitor processes	Displays interactive, auto-updating process statistics in a full-screen mode.

placement memory

To set the process memory threshold, use the **placement memory** command in global configuration or administration configuration mode. To return the settings to the default value, use the **no** form of this command.

placement memory {**maximum** | **threshold**} *value*

no placement memory

Syntax Description

maximum	Defines the maximum memory load level. The value is 10 percent to 2000 percent. The default value is 200 percent.
threshold	Defines the memory load level to trigger migration. The value is 10 percent to 400 percent. The default value is 80 percent.
value	Percent memory load value.

Defaults

The **maximum** value is 200 percent.
The **threshold** value is 80 percent.

Command Modes

Administration configuration
Global configuration

Command History

Release	Modification
Release 3.3.0	This command was introduced on the Cisco CRS-1 and Cisco XR 12000 Series Router.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

Use the **placement memory** command with the **maximum** *value* keyword and argument to set the maximum percentage of memory that can be used on a node (based on the estimated memory usage of the processes). For example:

- The command **placement memory maximum 100** sets the maximum memory usage on nodes to 100 percent. If a node has 2 GB of available memory, then only 2 GB of processes can be placed on the node (estimated memory usage). This means that oversubscription is not allowed.
- The command **placement memory maximum 50** sets the maximum memory usage on nodes to 50 percent, so that placeable processes can use only half the memory on the node.

- The command **placement memory maximum 200** would allow the system to attempt to run more than the available memory on a node.

Use the **placement memory** command with the **threshold** *value* keyword and argument to set the preferred percentage of memory use for each node. The system attempts to balance all nodes at or below the threshold memory percentage. In other words, the system does not place a process on a node that has exceeded the threshold value, unless all other nodes have also reached their thresholds (or unless some other large affinity overrides this consideration).

Use the **show placement policy global** command to display the current settings:

```
RP/0/RP0/CPU0:router# show placement policy global
Per-location placement policy parameters
----------------------------------------
Memory preferred threshold:      80%
Memory maximum threshold:       200%
Threshold satisfaction affinity points:    50
```

Task ID

Task ID	Operations
sysmgr	read, write

Examples

In the following example, the maximum memory threshold is set to 80 percent:

```
RP/0/RP0/CPU0:router(config)# placement memory maximum 80
```

Related Commands

Command	Description
show placement policy	Displays placement policy parameters and programs.

placement program

To enter placement program configuration mode to set process affinities (preferences), use the **placement program** command in global configuration mode. To remove the assigned process placement, use the **no** form of this command.

placement program {*program* [**instance** *instance*] | **default**}

no placement program *program* {**instance** *instance* | **default**}

Syntax Description

program	Process or group of processes.
instance *instance*	Specifies a program name that uniquely identifies a placement process. The process name is any alphanumeric string no longer than 40 characters.
default	Specifies all processes instead of a specific process instance.

Defaults

No default behavior or values

Command Modes

Global configuration

Command History

Release	Modification
Release 3.3.0	This command was supported on the Cisco CRS-1 and the Cisco XR 12000 Series Router.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

The **placement program** command allows you to change the placement policy for any placeable process, such as Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP).

Placement policy consists of two types: resource consumption and affinities. Resource consumption involves CPU and memory usage and is a “hard” resource requirement. Affinities are preferences, and are considered “soft” requirements for a placement policy. Only affinities can be configured by the operator.

The **placement program** command handles configuration for these affinity types:

- Affinity location set
- Affinity location type

- Affinity program
- Affinity self
- Affinity existence

To obtain a list of running placeable processes and placement policy parameters on your router system, use the **show placement policy** command.

Task ID

Task ID	Operations
sysmgr	read, write

Examples

The following example enters placement program configuration mode to set all Protocol Independent Multicast (PIM) processes on the node pair that it had already started on so that PIM does not move automatically when system conditions change:

```
RP/0/RP0/CPU0:router(config)# placement program pim
RP/0/RP0/CPU0:router(config-place)# affinity location-type current attract 100
```

Related Commands

Command	Description
affinity location-set	Sets the affinity of a placeable program (process) to or from node pairs.
affinity location-type	Sets the affinity of a placeable program (process) to or from a location type.
affinity program	Sets the affinity of a placeable program (process) to or from another program.
affinity self	Sets the affinity of a placeable program (process) to or from one of its own instances.
show placement location	Displays all placeable programs (processes) by location.
show placement policy	Displays placement policy parameters and programs.
show placement program	Displays the operational state for each placement program.

placement reoptimize

To reoptimize the placement of processes on a system among the available RP and DRP nodes in a secure domain router (SDR), use the **placement** command in EXEC mode.

placement reoptimize

Syntax Description

This command has no arguments or keywords.

Defaults

The default is to reoptimize process placement for all processes.

Command Modes

EXEC

Command History

Release	Modification
Release 3.3.0	This command was introduced on the Cisco CRS-1 and Cisco XR 12000 Series Router.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

Use the **placement reoptimize** command to reoptimize the placement of running proccesses on the available RP and DRP nodes. The processes are redistributed to the available nodes based on memory usage and other calculations. This **placement reoptimize** command first displays the predicted results of the reoptimization before running the command. You can accept the changes and run the command, or cancel the procedure without impacting the router.

Task ID

Task ID	Operations
sysmgr	read, write

Examples

The following example illustrates how to use the **placement reoptimize** command. The predicted changes are displayed, showing the current location of the process and the new location of the process. You can then decide to continue the operation or abort the reoptimization.

```
RP/0/RP0/CPU0:router# placement reoptimize

Predicted changes to the placement:
```

```
  bpm                 0/RP0/CPU0 (0/RP1/CPU0) --> 0/2/CPU0 (0/3/CPU0)
  bgp instance 0      0/RP0/CPU0 (0/RP1/CPU0) --> 0/2/CPU0 (0/3/CPU0)
  ipv4_rib            0/RP0/CPU0 (0/RP1/CPU0) --> 0/2/CPU0 (0/3/CPU0)
  ipv4_arm            0/RP0/CPU0 (0/RP1/CPU0) --> 0/2/CPU0 (0/3/CPU0)
  rcp_fs              0/RP0/CPU0 (0/RP1/CPU0) --> 0/2/CPU0 (0/3/CPU0)

Continue? [yes/no] yes
RP/0/RP0/CPU0:router#
RP/0/RP0/CPU0:Nov 12 1:1:1.1 : placed[170]: %PLACED_PLACE-6-
REOP_START: Re-optimization of the placement requested. You will be notified on
completion.
RP/0/RP0/CPU0:Nov 12 1:1:1.1 : placed[254]: %OS-PLACED_PLACE-6-REOP_COMPLETE
: Re-optimization of the placement complete. Use 'show placement' to view the ne
w placement
```

Related Commands

Command	Description
show placement reoptimize	Displays information about process placements by node or process.

process

To start, terminate, or restart a process, use the **process** command in administration EXEC mode.

process {**blocked | crash | restart | shutdown | start**} {*executable-name* | *job-id*} {**location** {*node-id* | **all**}}

Syntax Description

blocked	Collects debug information. The node is restarted if the mandatory reboot flag is set.
crash	Crashes a process.
restart	Restarts a process.
shutdown	Stops a process. The process is not restarted (even if considered "mandatory").
start	Starts a process.
executable-name	(Optional) Executable name of the process to be started, terminated, or restarted. Supplying an executable name for the *executable-name* argument performs the action for all the simultaneously running instances of the process, if applicable.
job-id	(Optional) Job ID of the process instance to be started, terminated, or restarted. Supplying a job ID for the *job-id* argument performs the action for only the process instance associated with the job ID.
location *node-id*	(Optional) Starts, terminates, or restarts a process on the designated node. The *node-id* argument is entered in the *rack*/*slot*/*module* notation.
location all	(Optional) Starts, terminates, or restarts a process on all nodes.

Defaults

No default behavior or values

Command Modes

Administration EXEC

Command History

Release	Modification
Release 2.0	This command was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	This command was first supported on the Cisco XR 12000 Series Router. The **shutdown** keyword was introduced to replace the **kill** keyword. Support for the **crash** keyword was added to crash a process.
Release 3.3.0	No modification.
Release 3.4.0	No modification.
Release 3.5.0	This command was removed from EXEC mode.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

Under normal circumstances, processes are started and restarted automatically by the operating system as required. If a process crashes, it is automatically restarted.

Use this command to manually stop, start, or restart individual processes.

Caution Manually stopping or restarting a process can seriously impact the operation of a router. Use these commands only under the direction of a Cisco Technical Support representative.

process shutdown

The **process shutdown** command shuts down (terminates) the specified process and copies associated with the specified process. The process is not restarted, even if considered "mandatory." Use the **show processes** command to display a list of executable processes running on the system.

Caution Stopping a process can result in an RP failover, system failure or both. This command is intended for use only under the direct supervision of a Cisco Technical Support representative.

process restart

The **process restart** command restarts a process, such as a process that is not functioning optimally.

process start

The **process start** command starts a process that is not currently running, such as a process that was terminated using the **process kill** command. If multiple copies are on the system, all instances of the process are started simultaneously.

process blocked

This command is used by Cisco Technical Support engineers to collect debug information about a process. If the **process mandatory** command is set for a process, the **process blocked** command also causes the node to restart.

Task ID

Task ID	Operations
root-lr	execute

Examples

The following example shows how to restart a process. In this example, the IS-IS process is restarted:

```
RP/0/RP0/CPU0:router# process restart isis

RP/0/RP0/CPU0:router#RP/0/RP0/CPU0:Mar 30 15:24:41 : isis[343]: %ISIS-6-INFO_ST
RTUP_START : Cisco NSF controlled start beginning
RP/0/RP0/CPU0:router#RP/0/RP0/CPU0:Mar 30 15:24:52 : isis[352]: %ISIS-6-INFO_ST
RTUP_FINISH : Cold controlled start completed
```

The following example shows how to terminate a process. In this example, the IS-IS process is stopped:

```
RP/0/RP0/CPU0:router# process shutdown isis
RP/0/RP0/CPU0:router#
```

The following example shows how to start a process. In this example, the IS-IS process is started:

```
RP/0/RP0/CPU0:router# process start isis

RP/0/RP0/CPU0:router#RP/0/RP0/CPU0:Mar 30 15:27:19 : isis[227]: %ISIS-6-INFO_STA
RTUP_START : Cold controlled start beginning
RP/0/RP0/CPU0:Mar 30 15:27:31 : isis[352]: %ISIS-6-INFO_STARTUP_FINISH : Cold co
ntrolled start completed
```

Related Commands

Command	Description
process mandatory	Sets the options for mandatory processes.
show processes	Displays information about the running processes.

process core

To modify the core dump options for a process, use the **process core** command in administration EXEC mode or in EXEC mode.

process {*executable-name* | *job-id*} **core** {**context** | **copy** | **fallback** | **iomem** | **mainmem** | **off** | **sharedmem** | **sparse** | **sync** | **text**} [**maxcore** *value*] **location** *node-id*

Syntax Description

executable-name	Executable name of the process for which you want to change core dump options. Specifying a value for the *executable-name* argument changes the core dump option for multiple instances of a running process.
job-id	Job ID associated with the process instance. Specifying a *job-id* value changes the core dump option for only a single instance of a running process.
context	Dumps only context information for a process.
copy	Copies a core dump locally before performing the core dump.
fallback	Sets the core dump options to use the fallback options (if needed).
iomem	Dumps the I/O memory of a process.
mainmem	Dumps the main memory of a process.
off	Indicates that a core dump is not taken on the termination of the specified process.
sharedmem	Dumps the shared memory of a process.
sparse	Enables sparse core dumps of a process.
sync	Enables only synchronous core dumping.
text	Dumps the text of a process.
maxcore *value*	(Optional) Specifies the maximum number of core dumps allowed for the specified process on its creation.
location *node-id*	Sets the core dump options for a process on a designated node. The *node-id* argument is entered in the *rack*/*slot*/*module* notation.

Defaults

By default, processes are configured to dump shared memory, text area, stack, data section, and heap information.

Command Modes

Administration EXEC
EXEC

Command History

Release	Modification
Release 2.0	This command was introduced on the Cisco CRS-1.
Release 3.0	No modification.

Release	Modification
Release 3.2	This command was first supported on the Cisco XR 12000 Series Router. The command was made available in administration EXEC mode. The **mainmem-sharedmem**, **mainmem-text**, **mainmem-text-sharedmem** keyword options were removed. The **context**, **fallback**, **iomem**, **sync**, and **text** keyword options were introduced.
Release 3.3.0	No modification.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

The modular architecture of the Cisco IOS XR software allows core dumps for individual processes. By default, processes are configured to dump shared memory, text area, stack, data section, and heap information.

Specifying an executable name for the *executable-name* argument changes the core dump option for all instances of the process. Specifying a job ID for the *job-id* value changes the core dump option for a single instance of a running process.

Task ID

Task ID	Operations
root-lr	execute

Examples

The following example shows how to enable the collection of shared memory of a process:

```
RP/0/RP0/CPU0:router# process ospf core sharedmem
```

The following example shows how to turn off core dumping for a process:

```
RP/0/RP0/CPU0:router# process media_ether_config_di core off
```

Related Commands

Command	Description
show processes	Displays information about the running processes.

process mandatory

To set the mandatory reboot options for a process, use the **process mandatory** command in administration EXEC mode or in EXEC mode.

process mandatory

process mandatory {**on** | **off**} {*executable-name* | *job-id*} **location** *node-id*

process mandatory reboot

process mandatory reboot {**enable** | **disable**}

process mandatory toggle

process mandatory toggle {*executable-name* | *job-id*} **location** *node-id*

Defaults

on	Turns on mandatory process attribute.
off	Turns off the mandatory process attribute. The process is not considered mandatory.
reboot {**enable** | **disable**}	Enables or disables the reboot action when a mandatory process fails.
toggle	Toggles a mandatory process attribute.
executable-name	Executable name of the process to be terminated. Specifying a executable name for the *executable-name* argument terminates the process and all the simultaneously running copies, if applicable.
job-id	Job ID associated with the process to be terminated. Terminates only the process associated with the job ID.
location *node-id*	(Optional) Sets the mandatory settings for a process on a designated node. The *node-id* argument is expressed in the *rack*/*slot*/*module* notation.

Command Default

No default behavior or values

Syntax Description

Administration EXEC
EXEC

Command History

Release	Modification
Release 2.0	This command was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	This command was first supported on the Cisco XR 12000 Series Router. The command was made available in administration EXEC mode.
Release 3.3.0	No modification.

Release	Modification
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

If a process unexpectedly goes down, the following action occurs based on whether the process is considered "mandatory."

- If the process is mandatory and the process cannot be restarted, the node automatically reboots.
- If the process is not mandatory and cannot be restarted, it stays down and the node does not reboot.

Task ID

Task ID	Operations
root-lr	execute

Examples

The following example shows how to turn on a mandatory attribute. In this example, the mandatory attribute is turned on for the media_ether_config_di process.

```
RP/0/RP0/CPU0:router# process mandatory on media_ether_config_di
```

The following example shows how to turn the reboot option on. In this example, the router is set to reboot the node if a mandatory process goes down and cannot be restarted.

```
RP/0/RP0/CPU0:router# process mandatory reboot enable

RP/0/RP0/CPU0:Mar 19 19:28:10 : sysmgr[71]: %SYSMGR-4-MANDATORY_REBOOT_ENABLE :
mandatory reboot option enabled by request
```

The following example shows how to turn off the reboot option. In this example, the router is set *not* to reboot the node if a mandatory process goes down and cannot be restarted. In this case, the mandatory process is restarted, but the node is not rebooted.

```
RP/0/RP0/CPU0:router# process mandatory reboot disable

RP/0/RP0/CPU0:Mar 19 19:31:20 : sysmgr[71]: %SYSMGR-4-MANDATORY_REBOOT_OVERRIDE
: mandatory reboot option overridden by request
```

Related Commands

Command	Description
show processes	Displays information about the running processes.

show context

To display core dump context information, use the **show context** command in administration EXEC mode or in EXEC mode.

show context [*coredump-occurrence* | **clear**] [**location** {*node-id* | **all**}]

Syntax Description

coredump-occurrence	(Optional) Core dump context information to be displayed based on the occurrence of the core dump. Valid values are 1 to 10. For example, entering a value of **1** for the *coredump-occurrence* argument displays core dump context information for the most recent core dump, while entering a value of **2** for the *coredump-occurrence* argument displays information for the second most recent core dump. This command stores context information for the last 10 core dumps.
clear	(Optional) Clears the current context information.
location {*node-id* | **all**}	(Optional) Displays core dump information that occurred on the designated node. The *node-id* argument is expressed in the *rack*/*slot*/*module* notation.

Command Default

If no *coredump-occurrence* value is specified, core dump context information for all core dumps is displayed.

Command Modes

Administration EXEC
EXEC

Command History

Release	Modification
Release 2.0	This command was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	This command was first supported on the Cisco XR 12000 Series Router. The command was made available in administration EXEC mode.
Release 3.3.0	No modification.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

Use the **show context** command to display core dump context information. This command displays context information for the last ten core dumps. Cisco Technical Support Center engineers and development engineers use this command for post-analysis in the debugging of processes.

Use the **clear context** command to clear core dump context information.

Task ID

Task ID	Operations
diag	read

Examples

The following is sample output from the **show context** command:

```
RP/0/RP0/CPU0:router# show context

Crashed pid = 20502 (pkg/bin/mbi-hello)
Crash time: Thu Mar 25, 2004: 19:34:14
Core for process at disk0:/mbi-hello.20040325-193414.node0_RP0_CPU0

    Stack Trace
#0 0xfc117c9c
#1 0xfc104348
#2 0xfc104154
#3 0xfc107578
#4 0xfc107734
#5 0x482009e4
               Registers info
            r0       r1       r2       r3
  R0   0000000e  481ffa80  4820c0b8  00000003
            r4       r5       r6       r7
  R4   481ffb18  00000001  481ffa88  48200434
            r8       r9      r10      r11
  R8   00000000  00000001  00000000  fc17ac58
           r12      r13      r14      r15
  R12  481ffb08  4820c080  481ffc10  00000001
           r16      r17      r18      r19
  R16  481ffc24  481ffc2c  481ffcb4  00000000
           r20      r21      r22      r23
  R20  00398020  00000000  481ffb6c  4820a484
           r24      r25      r26      r27
  R24  00000000  00000001  4820efe0  481ffb88
           r28      r29      r30      r31
  R28  00000001  481ffb18  4820ef08  00000001
           cnt       lr      msr       pc
  R32  fc168d58  fc104348  0000d932  fc117c9c
           cnd      xer
  R36  24000022  00000004

                      DLL Info
DLL path    Text addr.  Text size   Data addr. Data size   Version
/pkg/lib/libinfra.dll  0xfc0f6000  0x00032698  0xfc0f5268  0x00000cb4
```

The following is sample output from the **show context** command. The output displays information about a core dump from a process that has not crashed.

```
RP/0/RP0/CPU0:router# show context

node:      node0_RP0_CPU0
------------------------------------------------------------------

Crashed pid = 28703 (pkg/bin/packet)
```

```
Crash time: Tue Sep 21, 2004: 02:48:00
Core for process at harddisk:/packet.by.dumper_gen.20040921-024800.node0_RP0_CPU0.ppc.Z
```

Table 70 describes the significant fields shown in the display.

Table 70 show context Field Descriptions

Field	Description
Crashed pid =	Process ID (PID) of the crashed process followed by the executable path.
Crash time	Time and date the crash occurred.
Core for process at	File path to the core dump file.
Stack Trace	Stack trace information.
Registers Info	Register information related to crashed threads.
DLL Info	Dynamically loadable library (DLL) information used to decode the stack trace.

Related Commands

Command	Description
clear context	Clears core dump context information.

show dll

To display dynamically loadable library (DLL) information, use the **show dll** command in administration EXEC mode or in EXEC mode.

show dll [**jobid** *job-id* [**virtual**] | **address** *virtual-address* | **dllname** *dll-virtual-path* | **memory** | **symbol address** *virtual-address* | **virtual**] [**location** *node-id*]

Syntax Description

jobid *job-id* [**virtual**]	(Optional) Displays DLL information for the specified job identifier.
virtual	(Optional) Displays the virtual path of DLLs. The virtual path is expressed in the /pkg/lib/*library-name*.dll format where the *library-name* is the name of the DLL followed by the .dll suffix.
address *virtual-address*	(Optional) Displays the DLL that is mapped at the virtual address specified for the *virtual-address* argument.
dllname *dll-virtual-path*	(Optional) Displays the process IDs (PIDs) of the process that have downloaded the DLL specified for the *dll-virtual-path* argument.
memory	(Optional) Displays a summary of DLL memory usage.
symbol address *virtual-address*	(Optional) Displays the symbol at the virtual address specified for the *virtual-address* argument.
location *node-id*	(Optional) Displays DLLs for the specified node. The *node-id* argument is expressed in the *rack*/*slot*/*module* notation.

Command Default

No default behavior or values

Command Modes

Administration EXEC
EXEC

Command History

Release	Modification
Release 2.0	This command was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	This command was first supported on the Cisco XR 12000 Series Router. The command was made available in administration EXEC mode.
Release 3.3.0	No modification.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

Task ID

Task ID	Operations
basic-services	read

Examples

The following is sample output from the **show dll** command. In this example, the output displays all the DLLs loaded on the router:

```
RP/0/RP0/CPU0:router# show dll

DLL path                              Text VA    Text Sz    Data VA    Data Sz    Refcount
------------------------------------------------------------------------------------------
/lib/libui.dll                        0xfc000000 0x00007000 0xfc007000 0x00001000    1
/disk0/crs-base-0.48.0/lib/liblogin.dll 0xfc008000 0x00006000 0xfc00e000 0x00001000    1
/mbi/lib/libbanner.dll                0xfc00f000 0x00003000 0xfc012000 0x00001000    1
/disk0/crs-base-0.48.0/lib/libaaav2.dll 0xfc013000 0x0000f000 0xfc022000 0x00001000    1
/disk0/crs-base-0.48.0/lib/libaaatty.dll 0xfc023000 0x00004000 0xfc027000 0x00001000    1
/mbi/lib/libtermcap.dll               0xfc028000 0x00003000 0xfc02b000 0x00001000    1
/mbi/lib/lib_show_dll.dll             0xfc02c000 0x00004000 0xfc030000 0x00001000    1
/mbi/lib/libihplatform.dll            0xfc0bf2d4 0x00000c18 0xfc1e4f88 0x00000068    1
/lib/libovl.dll                       0xfc0c8000 0x0000c3b0 0xfc0c21f0 0x0000076c   23
/disk0/crs-admin-0.48.0/lib/libfqm_ltrace_util_common.dll 0xfc0d43b0 0x00000bfc 0xfc391f7c
0x00000068    1
/lib/libplatform.dll                  0xfc0d5000 0x0000aa88 0xfc0e0000 0x00002000  165
/lib/libsysmgr.dll                    0xfc0e2000 0x0000ab48 0xfc0c295c 0x00000368  166
/lib/libinfra.dll                     0xfc0ed000 0x0003284c 0xfc120000 0x00000c70  169
/lib/libios.dll                       0xfc121000 0x0002c4bc 0xfc14e000 0x00002000  166
/lib/libc.dll                         0xfc150000 0x00077ae0 0xfc1c8000 0x00002000  175
/mbi/lib/libltrace.dll                0xfc1ca000 0x00007f5c 0xfc0c2cc4 0x00000188   96
/lib/libsyslog.dll                    0xfc1d2000 0x0000530c 0xfc120c70 0x00000308  129
/disk0/crs-base-0.48.0/lib/liblpts_ifib_platform.dll 0xfc1d730c 0x00000cc8 0xfcef4000
0x00000068    1
/lib/libbackplane.dll                 0xfc1d8000 0x0000134c 0xfc0c2e4c 0x000000a8  163
/disk0/crs-base-0.48.0/lib/libipv6_platform_client.dll 0xfc1d934c 0x00000c48 0xfcef4f8c
0x00000068    1
/mbi/lib/libpkgfs_node.dll            0xfc1da000 0x000092d4 0xfc1e4000 0x000001a8    3
```

The following is sample output from the **show dll** command with the optional **jobid** keyword and *job-id* argument:

```
RP/0/RP0/CPU0:router# show dll jobid 186

DLLs mapped by PID 86111
DLL path                              Text VA    Text Sz    Data VA    Data Sz    Refcount
------------------------------------------------------------------------------------------
/lib/libovl.dll                       0xfc0c8000 0x0000c3b0 0xfc0c21f0 0x0000076c   23
/lib/libplatform.dll                  0xfc0d5000 0x0000aa88 0xfc0e0000 0x00002000  165
/lib/libsysmgr.dll                    0xfc0e2000 0x0000ab48 0xfc0c295c 0x00000368  167
/lib/libinfra.dll                     0xfc0ed000 0x0003284c 0xfc120000 0x00000c70  169
/lib/libios.dll                       0xfc121000 0x0002c4bc 0xfc14e000 0x00002000  166
/lib/libc.dll                         0xfc150000 0x00077ae0 0xfc1c8000 0x00002000  175
/mbi/lib/libltrace.dll                0xfc1ca000 0x00007f5c 0xfc0c2cc4 0x00000188   96
/lib/libsyslog.dll                    0xfc1d2000 0x0000530c 0xfc120c70 0x00000308  129
/lib/libbackplane.dll                 0xfc1d8000 0x0000134c 0xfc0c2e4c 0x000000a8  163
/lib/libnodeid.dll                    0xfc1e5000 0x000091fc 0xfc1e41a8 0x00000208  163
/mbi/lib/libinst_mem.dll              0xfc232000 0x000044f8 0xfc1e43b0 0x00000108    4
/lib/libdebug.dll                     0xfc23c000 0x0000ef64 0xfc1e4680 0x00000550  159
```

Table 71 describes the significant fields shown in the display.

Table 71 show dll Field Descriptions

Field	Description
DLL path	Physical path of the DLL on the router.
Text VA	Virtual address of the text segment of the DLL.
Text Sz	Size of the text segment of the DLL.
Data VA	Virtual address of the data segment of the DLL.
Data Sz	Size of the data segment of the DLL.
Refcount	Count of the number of clients using the DLL.

The following is sample output from the **show dll** command with the optional **dllname** keyword and optional *dll-virtual-path* argument:

```
RP/0/RP0/CPU0:router# show dll dllname /pkg/lib/libinst_mem.dll

  PID:     4102  Refcount: 1
  PID:     4105  Refcount: 1
  PID:    24600  Refcount: 1
  PID:    86111  Refcount: 1
```

Table 72 describes the significant fields shown in the display.

Table 72 show dll dllname Field Descriptions

Field	Description
PID:	Process ID of the process.
Refcount	Number of references to the DLL by the process.

The following is sample output from the **show dll** command with the optional **memory** keyword:

```
RP/0/RP0/CPU0:router# show dll memory
------------------------------------------------------------------------------

Total DLL Text - 14778896 bytes  Total DLL Data - 12688500 bytes
Total DLL Memory - 27467396 bytes
```

show exception

To display the configured core dump settings, use the **show exception** command in administration EXEC mode or in EXEC mode.

show exception

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Administration EXEC
EXEC

Command History

Release	Modification
Release 2.0	This command was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	This command was first supported on the Cisco XR 12000 Series Router.
Release 3.3.0	No modification.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

Use the **show exception** command to display the configured core dump settings. The output from this command displays the core dump settings configured with the following commands:

- **exception filepath**
- **exception pakmem**
- **exception sparse**
- **exception sprsize**

Task ID

Task ID	Operations
diag	read

Examples

The following is sample output from the **show exception** command:

```
RP/0/RP0/CPU0:router# show exception

 Choice  1 path = harddisk:/coredump compress = on filename = <process_name.time>
 Choice  2 path = tftp://223.255.254.254/users/xyz compress = on filename =
<process_name.time>
 Exception path for choice 3 is not configured or removed
 Choice fallback one path = harddisk:/dumper compress = on filename = <process_name>
 Choice fallback two path = disk1:/dumper compress = on filename = <process_name>
 Choice fallback three path = disk0:/dumper compress = on filename = <process_name>
 Kernel dump not configured
 Tftp route for kernel core dump not configured
 Dumper packet memory in core dump enabled
 Sparse core dump enabled
 Dumper will switch to sparse core dump automatically at size 300MB
```

Related Commands

Command	Description
exception filepath	Modifies core dump settings.
exception pakmem	Collects packet memory information in core dumps.
exception sparse	Enables or disables sparse core dumps.
exception sprsize	Sets the maximum size of core dump files.

show memory

To display the available physical memory and memory usage information of processes on the router, use the **show memory** command in administration EXEC mode or in EXEC mode.

show memory [*job-id* | **summary** [**bytes** | **detail**]] **location** *node-id*

Syntax Description

job-id	(Optional) Job ID associated with a process instance. Specifying a job ID for the *job-id* argument displays the memory available and memory usage information for only the process associated with the specified job ID. If the *job-id* argument is not specified, this command displays information for all running processes.
summary	(Optional) Displays a summary of the physical memory and memory usage information.
bytes	(Optional) Displays numbers in bytes for an exact count.
detail	(Optional) Displays numbers in the format "nnn.dddM" for more detail.
location *node-id*	Displays the available physical memory from the designated node. The *node-id* argument is entered in the *rack*/*slot*/*module* notation.

Command Default

No default behavior or values

Command Modes

Administration EXEC
EXEC

Command History

Release	Modification
Release 2.0	This command was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	This command was first supported on the Cisco XR 12000 Series Router. The command was made available in administration EXEC mode.
Release 3.3.0	No modification.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

To display detailed memory information for the entire router, enter the **show memory** command without any parameters.

Task ID

Task ID	Operations
basic-services	read

Examples

The following is partial sample output from the **show memory** command entered without keywords or arguments. This command displays details for the entire router.

```
RP/0/RP0/CPU0:router# show memory

Physical Memory:2048M total
 Application Memory :1802M (1636M available)
 Image:116M (bootram:116M)
 Reserved:128M, IOMem:0, flashfsys:0
 Total shared window:0

kernel:jid 1
Address         Bytes           What
0008f000        12288           Program Stack
000b2000        12288           Program Stack
Total Allocated Memory:0
Total Shared Memory:0

sbin/devc-pty:jid 68
Address         Bytes           What
4817f000        4096            Program Stack (pages not allocated)
48180000        516096          Program Stack (pages not allocated)
481fe000        8192            Program Stack
48200000        28672           Physical Mapped Memory
48207000        4096            ANON FIXED ELF SYSRAM
48208000        4096            ANON FIXED ELF SYSRAM
```

The following is sample output from the **show memory** command entered with the job ID 7 to show the memory usage information for the process associated with this job identifier:

```
RP/0/RP0/CPU0:router# show memory 7

Physical Memory: 256M total
 Application Memory : 249M (217M available)
 Image: 2M (bootram: 2M)
 Reserved: 4M, IOMem: 0, flashfsys: 0

sbin/pipe: jid 7
Address         Bytes           What
07f7c000        126976          Program Stack (pages not allocated)
07f9b000        4096            Program Stack
07f9d000        126976          Program Stack (pages not allocated)
07fbc000        4096            Program Stack
07fbe000        126976          Program Stack (pages not allocated)
07fdd000        4096            Program Stack
07fdf000        126976          Program Stack (pages not allocated)
07ffe000        4096            Program Stack
08000000        122880          Program Stack (pages not allocated)
0801e000        8192            Program Stack
08020000        12288           Physical Mapped Memory
08023000        4096            Program Text or Data
08024000        4096            Program Text or Data
08025000        16384           Allocated Memory
08029000        16384           Allocated Memory
7c001000        319488          DLL Text libc.dll
7e000000        8192            DLL Data libc.dll
```

The following example shows how to display a detailed summary of memory information for the router:

```
RP/0/0/CPU0:router# show memory summary detail

Physical Memory: 256.000M total
 Application Memory : 140.178M (15.003M available)
 Image: 95.739M (bootram: 95.739M)
 Reserved: 20.000M, IOMem: 0, flashfsys: 0
 Shared window fibv6: 257.980K
 Shared window PFI_IFH: 207.925K
 Shared window aib: 8.972M
 Shared window infra_statsd: 3.980K
 Shared window ipv4_fib: 1.300M
 Shared window atc_cache: 35.937K
 Shared window qad: 39.621K
 Total shared window: 10.805M
 Allocated Memory: 49.933M
 Program Text: 6.578M
 Program Data: 636.000K
 Program Stack: 4.781M
```

Table 73 describes the significant fields shown in the display.

Table 73 show memory summary Field Descriptions

Field	Description
Physical Memory	Available physical memory on the router.
Application Memory	Current memory usage of all the processes on the router.
Image	Memory that is currently used by the image, and available memory.
Reserved	Total reserved memory.
IOMem	Available I/O memory.
flashfsys	Total flash memory.
Shared window fibv6	Internal shared window information.
Shared window PFI_IFH	Internal shared window information.
Shared window aib	Internal shared window information.
Shared window infra_statsd	Internal shared window information.
Shared window ipv4_fib	Internal shared window information.
Shared window atc_cache	Internal shared window information.
Shared window qad	Internal shared window information.
Total shared window	Internal shared window information.
Allocated Memory	Amount of memory allocated for the specified node.
Program Text	Internal program test information.
Program Data	Internal program data information.
Program Stack	Internal program stack information.

Related Commands

Command	Description
show memory heap	Displays information about the heap space for a process.
show processes	Displays information about the running processes.

show memory compare

To display details about heap memory usage for all processes on the router at different moments in time and compare the results, use the **show memory compare** command in administration EXEC mode or in EXEC mode.

show memory compare {**start** | **end** | **report**}

Syntax Description

start	Takes the initial snapshot of heap memory usage for all processes on the router and sends the report to a temporary file named /tmp/memcmp_start.out.
end	Takes the second snapshot of heap memory usage for all processes on the router and sends the report to a temporary file named /tmp/memcmp_end.out. This snapshot is compared with the initial snapshot when displaying the heap memory usage comparison report.
report	Displays the heap memory comparison report, comparing heap memory usage between the two snapshots of heap memory usage.

Command Default

No default behavior or values

Command Modes

Administration EXEC
EXEC

Command History

Release	Modification
Release 2.0	This command was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	This command was first supported on the Cisco XR 12000 Series Router. The command was made available in administration EXEC mode.
Release 3.3.0	No modification.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

Use the **show memory compare** command to display details about the heap memory usage of all processes on the router at different moments in time and compare the results. This command is useful for detecting patterns of memory usage during events such as restarting processes or configuring interfaces.

Use the following steps to create and compare memory snapshots:

Step 1 Enter the **show memory compare start** command to take the initial snapshot of heap memory usage for all processes on the router.

Note The snapshot is similar to that resulting from entry of the **show memory heap** command with the optional **summary** keyword.

Step 2 Perform the test you want to analyze.

Step 3 Enter the **show memory compare end** command to take the snapshot of heap memory usage to be compared with the initial snapshot.

Step 4 Enter the **show memory compare report** command to display the heap memory usage comparison report.

Task ID

Task ID	Operations
basic-services	read

Examples

The following is sample output from the **show memory compare report** command:

```
RP/0/RP0/CPU0:router# show memory compare report

JID   name                 mem before   mem after    difference   mallocs  restarted
---   ----                 ----------   ---------    ----------   -------  ---------
84    driver_infra_partner 577828       661492       83664        65
279   gsp                  268092       335060       66968        396
236   snap_transport       39816        80816        41000        5
237   mpls_lsd_agent       36340        77340        41000        5
268   fint_partner         24704        65704        41000        5
90    null_caps_partner    25676        66676        41000        5
208   aib                  55320        96320        41000        5
209   ipv4_io              119724       160724       41000        5
103   loopback_caps_partne 33000        74000        41000        5
190   ipv4_arm             41432        82432        41000        5
191   ipv6_arm             33452        74452        41000        5
104   sysldr               152164       193164       41000        5
85    nd_partner           37200        78200        41000        5
221   clns                 61520        102520       41000        5
196   parser_server        1295440      1336440      41000        5
75    bundlemgr_distrib    57424        98424        41000        5
200   arp                  83720        124720       41000        5
201   cdp                  56524        97524        41000        5
204   ether_caps_partner   39620        80620        41000        5
206   qosmgr               55624        96624        41000        5
240   imd_server           92880        104680       11800        28
260   improxy              77508        88644        11136        10
111   nrssvr               29152        37232        8080         60
275   sysdb_svr_local      1575532      1579056      3524         30
205   cfgmgr               31724        33548        1824         25
99    sysdb_svr_shared     1131188      1132868      1680         14
51    mbus-rp              26712        27864        1152         4
66    wdsysmon             298068       299216       1148         15
```

```
168    netio                1010912      1012060      1148         6
283    itrace_manager       17408        17928        520          3
59     devc-conaux          109868       110300       432          4
67     syslogd_helper       289200       289416       216          2
117    fctl                 41596        41656        60           2
54     sysmgr               171772       171076       -696         -5
269    ifmgr                539308       530652       -8656        -196        *
```

Table 74 describes the significant fields shown in the display.

Table 74 show memory compare report Field Descriptions

Field	Description
JID	Process job ID.
name	Process name.
mem before	Heap memory usage at start (in bytes).
mem after	Heap memory usage at end (in bytes).
difference	Difference in heap memory usage (in bytes).
mallocs	Number of unfreed allocations made during the test period.
restarted	Indicates if the process was restarted during the test period.

Related Commands

Command	Description
show memory heap	Displays information about the heap space for a process.
show processes	Displays information about the running processes.

show memory heap

To display information about the heap space for a process, use the **show memory heap** command in administration EXEC mode or in EXEC mode.

show memory heap {*job-id* | **allocated** [**dllname**] [**free**] *job-id* | **dllname** *job-id* | **free** [**dllname**] *job-id* | **failure** {*job-id* | **all**} | **summary** {*job-id* | **all**}}

Syntax Description

job-id	Job ID associated with the process instance.
allocated	(Optional) Displays a list of all allocated heap blocks.
dllname	(Optional) Displays heaps with dynamic link library (DLL) names.
failure	(Optional) Displays a summary of heap failures.
free	(Optional) Displays a list of all free heap blocks.
summary	(Optional) Displays a summary of the information about the heap space.
all	Displays information about the heap space for all processes.

Defaults

No default behavior or values

Command Modes

Administration EXEC
EXEC

Command History

Release	Modification
Release 2.0	This command was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	This command was first supported on the Cisco XR 12000 Series Router. The command was made available in administration EXEC mode.
Release 3.3.0	No modification.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

Task ID

Task ID	Operations
basic-services	read

Examples

The following is sample output from the **show memory heap** command, specifying a job ID for the *job-id* argument:

```
RP/0/RP0/CPU0:router# show memory heap 111

Malloc summary for pid 16433:
  Heapsize 16384: allocd 6328, free 8820, overhead 1236
  Calls: mallocs 144; reallocs 73; frees 5; [core-allocs 1; core-frees 0]
Block Allocated List
Total      Total      Block       Name/ID/Caller
Usize      Size       Count

0x000008c1 0x000008cc 0x00000001  0x7c018a10
0x000005ac 0x00000974 0x00000079  0x7c02b9e0
0x000004f0 0x000004f8 0x00000001  0x7c02b6fc
0x00000080 0x00000088 0x00000001  0x7c01936c
0x00000034 0x00000048 0x00000001  0x7c018954
0x00000024 0x00000030 0x00000001  0x7c019278
0x00000018 0x00000020 0x00000001  0x7c019b2c
0x00000008 0x00000010 0x00000001  0x7c017178
0x00000008 0x00000010 0x00000001  0x7c00fb54
0x00000008 0x00000010 0x00000001  0x7c00fb80
0x00000008 0x00000010 0x00000001  0x7c00fbb8
```

Table 75 describes the significant fields shown in the display.

Table 75 show memory heap Field Descriptions

Field	Description
Malloc summary for pid	System-defined process ID (PID).
Heapsize	Size of the heap as allocated from the system by the malloc library.
allocd	Bytes allocated to the process.
free	Bytes available in the heap.
overhead	Malloc library overhead in bytes.
mallocs	Number of malloc calls.
reallocs	Number of realloc calls.
frees	Number of invocations to the caller interface provided in the malloc library for deallocating the memory.
[core-allocs 1; core-frees 0]	Number of core memory units, the memory units in the malloc library allocated by the system for the heap, allocated and freed.

The following is sample output from the **show memory heap** command, specifying the **summary** keyword and a job ID for the *job-id* argument:

```
RP/0/RP0/CPU0:router# show memory heap summary 65

Malloc summary for pid 20495 process pcmciad:
  Heapsize 65536: allocd 40332, free 16568, overhead 8636
  Calls: mallocs 883; reallocs 3; frees 671; [core-allocs 4; core-frees 0]
Band size 16, element per block 48, nbuint 1
  Completely free blocks: 0
  Block alloced: 2, Block freed: 0
  allocs: 85, frees: 20
```

```
  allocmem: 1040, freemem: 496, overhead: 448
  blocks: 2, blknodes: 96
Band size 24, element per block 34, nbuint 1
  Completely free blocks: 0
  Block alloced: 1, Block freed: 0
  allocs: 243, frees: 223
  allocmem: 480, freemem: 336, overhead: 168
  blocks: 1, blknodes: 34
Band size 32, element per block 26, nbuint 1
  Completely free blocks: 0
  Block alloced: 1, Block freed: 0
  allocs: 107, frees: 97
  allocmem: 320, freemem: 512, overhead: 136
  blocks: 1, blknodes: 26
Band size 40, element per block 22, nbuint 1
  Completely free blocks: 0
  Block alloced: 2, Block freed: 0
  allocs: 98, frees: 74
  allocmem: 960, freemem: 800, overhead: 240
  blocks: 2, blknodes: 44
Band size 48, element per block 18, nbuint 1
  Completely free blocks: 0
  Block alloced: 1, Block freed: 0
  allocs: 53, frees: 42
  allocmem: 528, freemem: 336, overhead: 104
  blocks: 1, blknodes: 18
Band size 56, element per block 16, nbuint 1
  Completely free blocks: 0
  Block alloced: 1, Block freed: 0
  allocs: 8, frees: 4
  allocmem: 224, freemem: 672, overhead: 96
  blocks: 1, blknodes: 16
Band size 64, element per block 14, nbuint 1
  Completely free blocks: 0
  Block alloced: 1, Block freed: 0
  allocs: 6, frees: 2
  allocmem: 256, freemem: 640, overhead: 88
  blocks: 1, blknodes: 14
Band size 72, element per block 12, nbuint 1
  Completely free blocks: 0
  Block alloced: 1, Block freed: 0
  allocs: 1, frees: 0
  allocmem: 72, freemem: 792, overhead: 80
  blocks: 1, blknodes: 12
```

Table 76 describes the significant fields shown in the display.

Table 76 show memory heap summary Field Descriptions

Field	Description
Malloc summary for pid	System defined process ID (pid).
Heapsize	Size of the heap as allocated from the system by the malloc library.
allocd	Bytes allocated to the process.
free	Bytes available in the heap.
overhead	Malloc library overhead in bytes.
mallocs	Number of malloc calls.
reallocs	Number of realloc calls.

Table 76 show memory heap summary Field Descriptions (continued)

Field	Description
frees	Number of invocations to the caller interface provided in the malloc library for deallocating the memory.
[core-allocs 1; core-frees 0]	Number of core memory units, the memory units in the malloc library allocated by the system for the heap, allocated and freed.
Band size	Small memory elements are arranged in bands. The band size specifies the size of elements within the band.
element per block	Number of elements per block in the band.
nbunit	Number of memory unit one block consists of. Any block in any band should be of a size that is an integer multiple of this basic unit.
Completely free blocks	Number of blocks in the band completely free (available for allocation).
Block alloced	Number of blocks currently allocated for the band.
allocs	Number of allocations currently performed from the band.
frees	Number of free calls that resulted in memory being returned to the band.
allocmem	Amount of memory currently allocated from the band.
overhead	Amount of memory in bytes as overhead for managing the band.
blocks	Number of blocks currently in the band.
blknodes	Number of nodes (elements) in all the blocks in the band.

Related Commands

Command	Description
show memory	Displays the available physical memory and processes memory on a router.

show placement location

To display all placeable processes by location, use the **show placement location** command in EXEC mode.

show placement location {*node-id* | **all**}

Syntax Description

node-id	The *node-id* argument is expressed in the *rack*/*slot*/*module* notation.
all	Displays all placement locations.

Defaults

No default behavior or values

Command Modes

EXEC

Command History

Release	Modification
Release 3.3.0	This command was introduced on the Cisco CRS-1.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

The **show placement location** command identifies the process placement locations, the operational state of the nodes, and the processes currently running on the nodes.

To display the node location and operational state for a program, use the **show placement program** command.

Task ID

Task ID	Operations
sysmgr	read

Examples

The following example shows all the nodes on the routing system and the placeable programs on those node locations:

```
RP/0/RP0/CPU0:router# show placement location all

Locations: [0/RP0/CPU0] 0/RP1/CPU0 (RP)
State: running   Primary: yes   Paired: no
```

```
Programs placed at this location:
  rt_check_mgr
  fm_server
  fm_script_dir
  fm_metric_dir
  fm_fd_stats
  fm_fd_hardware
  fm_fd_drvinfra
  fm_fd_counter
  ipv6_rump
  ipv6_local
  ipv6_connected
  ipv4_rump
  ipv4_local
  ipv4_connected
  tftp_fs
  rcp_fs
  ftp_fs
  domain_services
  bfd
  ipv6_mpa
  ipv4_mpa
  ipv6_arm
  ipv4_arm
  policy_repository
  ipv6_rib
  ipv4_rib
  cdp_mgr
  statsd_manager
```

Table 77 describes the significant fields shown in the display.

Table 77 show placement location Field Descriptions

Field	Description
Locations	Physical placement locations identified on the routing system.
State	Operational state of the nodes.
Primary	Whether or not the locations are primary nodes.
Paired	Whether or not the locations are node pairs (active and standby).
Programs placed at this location	All processes that are currently placed at the location.

Related Commands

Command	Description
show placement program	Displays the operational state for each placement program.

show placement policy

To display placement policy parameters and programs, use the **show placement policy** command in EXEC mode.

show placement policy {**global** | **program** {*program* | **all** | **default**}}

Syntax Description

global	Displays system-wide placement policies.
program	Displays program placement policies.
program	Specific program or program group.
all	Displays all program placement policies.
default	Displays the default placement policies.

Command Default

No default behavior or values

Command Modes

EXEC

Command History

Release	Modification
Release 3.3.0	This command was introduced on the Cisco CRS-1.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

The **show placement policy** command displays placement policy items that influence program or process placement. The **global** option lists the per node threshold parameters, and the remaining options list the policy based on program.

If you configured the placement policy, the scope could be the default program, a given program, or even a particular program instance. Alternatively, the policy could be specified in a placement file (entered by the system) with different levels of affinities. To view all this information, you can specify the **program all** option to display the source of each program, and the policy applied to the default program or the program that you configured to override it.

Task ID

Task ID	Operations
sysmgr	read

Examples

The following example shows how to display the global parameters for the placement policy:

```
RP/0/RP0/CPU0:router# show placement policy global

Per-location placement policy parameters
----------------------------------------
Memory preferred threshold:      80%
Memory maximum threshold:       200%
Threshold satisfaction affinity points:    50
```

Table 78 describes the significant fields shown in the display.

Table 78 show placement policy global Field Descriptions

Field	Description
Memory preferred threshold	Memory threshold that should not be exceeded when placing processes.
Memory maximum threshold	Hard-memory threshold that should not be exceeded when placing processes.
Threshold satisfaction affinity points	Number of affinity points awarded to a node with empty CPU or memory. Points are lowered as CPU or memory is used, reaching zero when the threshold value is reached.

The following example displays placement policy for all programs running on the system:

```
RP/0/RP0/CPU0:router# show placement policy program all

Program: rsvp                                        : source
-------------------------------------------------------------
  Assumed mem:                1 MB                   : system [default]
  Slow migration interval:    1 second               : system [default]

  affinity location-type primary attract 40          : system [default]
  affinity location-type paired attract 60           : system [default]
  affinity existence attract 90                      : system [default]
  affinity location-type current attract 100         : system [default]

Program: [default]                                   : source
-------------------------------------------------------------
  Assumed mem:                1 MB                   : system [default]
  Slow migration interval:    1 second               : system [default]

  affinity location-type primary attract 40          : system [default]
  affinity location-type paired attract 60           : system [default]
  affinity existence attract 90                      : system [default]
  affinity location-type current attract 100         : system [default]

Program: mpls_static                                 : source
-------------------------------------------------------------
  Assumed mem:                1 MB                   : system [default]
  Slow migration interval:    1 second               : system [default]

  affinity location-type primary attract 40          : system [default]
  affinity location-type paired attract 60           : system [default]
  affinity existence attract 90                      : system [default]
  affinity location-type current attract 100         : system [default]
```

```
Program: mpls_rid_helper                              : source
------------------------------------------------------------
  Assumed mem:                1 MB                    : system [default]
  Slow migration interval:    1 second                : system [default]

  affinity location-type primary attract 40           : system [default]
  affinity location-type paired attract 60            : system [default]
  affinity existence attract 90                       : system [default]
  affinity location-type current attract 100          : system [default]

Program: mpls_ldp                                     : source
------------------------------------------------------------
  Assumed mem:                1 MB                    : system [default]
  Slow migration interval:    1 second                : system [default]

  affinity location-type primary attract 40           : system [default]
  affinity location-type paired attract 60            : system [default]
  affinity existence attract 90                       : system [default]
  affinity location-type current attract 100          : system [default]

Program: isis                                         : source
------------------------------------------------------------
  Assumed mem:                1 MB                    : system [default]
  Slow migration interval:    1 second                : system [default]

  affinity location-type primary repulse 40           : system isis
  affinity location-type paired attract 60            : system [default]
  affinity existence attract 90                       : system [default]
  affinity location-type current attract 100          : system [default]

Program: ipv6_static                                  : source
------------------------------------------------------------
  Assumed mem:                1 MB                    : system [default]
  Slow migration interval:    1 second                : system [default]

  affinity location-type primary repulse 40           : system ipv6_static
  affinity location-type paired attract 60            : system [default]
  affinity existence attract 90                       : system [default]
  affinity location-type current attract 100          : system [default]

Program: ipv4_static                                  : source
------------------------------------------------------------
  Assumed mem:                1 MB                    : system [default]
  Slow migration interval:    1 second                : system [default]

  affinity location-type primary repulse 40           : system ipv4_static
  affinity location-type paired attract 60            : system [default]
  affinity existence attract 90                       : system [default]
  affinity location-type current attract 100          : system [default]

Program: brib                                         : source
------------------------------------------------------------
  Assumed mem:                1 MB                    : system [default]
  Slow migration interval:    1 second                : system [default]

  affinity location-type current attract 600          : system brib
  affinity location-type paired attract 200           : system brib
  affinity location-type primary repulse 150          : system brib
  affinity program ipv6_rib attract 70                : system brib
  affinity program ipv4_rib attract 70                : system brib
  affinity existence attract 90                       : system [default]
```

```
Program: ipv6_rib                                       : source
-------------------------------------------------------------
  Assumed mem:                1 MB                      : system [default]
  Slow migration interval:    1 second                  : system [default]

  affinity location-type paired attract 250             : system ipv6_rib
  affinity location-type primary repulse 200            : system ipv6_rib
  affinity program brib attract 70                      : system ipv6_rib
  affinity program bgp attract 250                      : system ipv6_rib
  affinity existence attract 90                         : system [default]
  affinity location-type current attract 100            : system [default]

Program: ipv4_rib                                       : source
-------------------------------------------------------------
  Assumed mem:                1 MB                      : system [default]
  Slow migration interval:    1 second                  : system [default]

  affinity location-type paired attract 250             : system ipv4_rib
  affinity location-type primary repulse 200            : system ipv4_rib
  affinity program brib attract 70                      : system ipv4_rib
  affinity program bgp attract 250                      : system ipv4_rib
  affinity existence attract 90                         : system [default]
  affinity location-type current attract 100            : system [default]

Program: bgp                                            : source
-------------------------------------------------------------
  Assumed mem:                1 MB                      : system [default]
  Slow migration interval:    1 second                  : system [default]

  affinity location-type current attract 600            : system bgp
  affinity location-type paired attract 50              : system bgp
  affinity location-type primary repulse 50             : system bgp
  affinity self repulse 160                             : system bgp
  affinity program ipv6_rib attract 250                 : system bgp
  affinity program ipv4_rib attract 250                 : system bgp
  affinity existence attract 90                         : system [default]
```

Table 79 describes the significant fields shown in the display.

Table 79 show placement policy program all Field Descriptions

Field	Description
program	Placement policy program.
source	Position in the hierarchy for the policy origination: default entity, process class, or specific instance.
Assumed mem	Memory usage value from configuration or program placement file; defaults to 1 MB.

show placement program

To display the operational state for each placement program, use the **show placement program** command in EXEC mode.

show placement program {*program* | **all**}

Syntax Description

program	Specific program or program group.
all	Displays operational state for all placement programs.

Defaults

No default behavior or values

Command Modes

EXEC

Command History

Release	Modification
Release 3.3.0	This command was introduced on the Cisco CRS-1.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

The **show placement program** command displays information that identifies that the process (or program) is running, where the process is located, locations where the process was rejected, and location where the process is waiting to start.

Task ID

Task ID	Operations
sysmgr	read

Examples

The following example shows that all placement programs are running on the node pair 0/RP0/CPU0 and 0/RP1/CPU0:

```
RP/0/RP0/CPU0:router# show placement program all

If a program is shown as having 'rejected locations' (i.e., locations on which
it cannot be placed), the locations in question can been seen using the "show
placement policy program" command.

If a program has been placed but not yet started, the amount of time elapsed
```

```
since the program was placed is shown in the 'waiting to start' field.

Parentheses around the node indicate that the node has not yet fully booted.
  This will be true of standby nodes.

Program                        Placed at location            # rejected  Waiting
                                                             locations   to start
-------------------------------------------------------------------------------
mpls_vpn_mib                   0/RP0/CPU0 (0/RP1/CPU0)
ipv4_static                    0/RP0/CPU0 (0/RP1/CPU0)
rt_check_mgr                   0/RP0/CPU0 (0/RP1/CPU0)
mpls_rid_helper                0/RP0/CPU0 (0/RP1/CPU0)
ital_test2                     0/RP0/CPU0 (0/RP1/CPU0)
ital_test1                     0/RP0/CPU0 (0/RP1/CPU0)
fm_metric_dir                  0/RP0/CPU0 (0/RP1/CPU0)
fm_fd_stats                    0/RP0/CPU0 (0/RP1/CPU0)
fm_fd_hardware                 0/RP0/CPU0 (0/RP1/CPU0)
fm_fd_drvinfra                 0/RP0/CPU0 (0/RP1/CPU0)
fm_fd_counter                  0/RP0/CPU0 (0/RP1/CPU0)
ipv6_rump                      0/RP0/CPU0 (0/RP1/CPU0)
ipv6_local                     0/RP0/CPU0 (0/RP1/CPU0)
ipv6_connected                 0/RP0/CPU0 (0/RP1/CPU0)
ipv4_rump                      0/RP0/CPU0 (0/RP1/CPU0)
ipv4_local                     0/RP0/CPU0 (0/RP1/CPU0)
ipv4_connected                 0/RP0/CPU0 (0/RP1/CPU0)
tftp_fs                        0/RP0/CPU0 (0/RP1/CPU0)
rcp_fs                         0/RP0/CPU0 (0/RP1/CPU0)
ftp_fs                         0/RP0/CPU0 (0/RP1/CPU0)
domain_services                0/RP0/CPU0 (0/RP1/CPU0)
bfd                            0/RP0/CPU0 (0/RP1/CPU0)
ipv6_mpa                       0/RP0/CPU0 (0/RP1/CPU0)
ipv4_mpa                       0/RP0/CPU0 (0/RP1/CPU0)
ipv6_arm                       0/RP0/CPU0 (0/RP1/CPU0)
ipv4_arm                       0/RP0/CPU0 (0/RP1/CPU0)
policy_repository              0/RP0/CPU0 (0/RP1/CPU0)
ipv6_rib                       0/RP0/CPU0 (0/RP1/CPU0)
ipv4_rib                       0/RP0/CPU0 (0/RP1/CPU0)
statsd_manager                 0/RP0/CPU0 (0/RP1/CPU0)
rsi_master                     0/RP0/CPU0 (0/RP1/CPU0)
RP/0/RP0/CPU0:router#
```

Table 80 describes the significant fields shown in the display.

Table 80 show placement program Field Descriptions

Field	Description
Program	Name of the program that is placed.
Placed at location	Location of the RP at which this process is placed along with the paired node information.
rejected locations	Lists of nodes from which the program was rejected.
Waiting to start	The program is waiting in the process placement queue.

Related Commands

Command	Description
show placement policy	Displays placement policy parameters and programs.

show placement reoptimize

To display the predicted changes to reoptimize the placement of processes, use the **show placement reoptimize** command in EXEC mode.

show placement reoptimize [**program** {*program* **instance** *program-instance* | **all**}]

Syntax Description

program	(Optional) Displays changes for a particular program.
program	A specific program or program group.
instance program-instance	A specific program within a program group.
all	Displays changes for all programs.

Defaults

The default is to display predicted changes for all for all route processor (RP) and distributed route processor (DRP) nodes and processes.

Command Modes

EXEC

Command History

Release	Modification
Release 3.3.0	This command was introduced on the Cisco CRS-1 and Cisco XR 12000 Series Router.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

Use the **show placement reoptimize** command to display the predicted changes that will occur when the **placement reoptimize** command is run.

Task ID

Task ID	Operations
sysmgr	read

Examples

The following example is of the **show placement reoptimize** command, entered without keywords or arguments:

```
RP/0/RP0/CPU0:router# show placement reoptimize
```

```
Predicted changes to the placement

Program                 Current location             New location
-------------------------------------------------------------------------------
ipv4_rib                0/7/CPU1 (1/14/CPU1)         1/RP0/CPU0 (1/RP1/CPU0)
ipv6_rib                0/7/CPU1 (1/14/CPU1)         1/RP0/CPU0 (1/RP1/CPU0)
brib instance 81        0/7/CPU1 (1/14/CPU1)         1/RP0/CPU0 (1/RP1/CPU0)
bgp instance 1          0/7/CPU1 (1/14/CPU1)         1/RP0/CPU0 (1/RP1/CPU0)
```

Related Commands

Command	Description
placement reoptimize	Reoptimizes the processes among the available RPs and DRPs.

show processes

To display information about active processes, use the **show processes** command in administration EXEC or in EXEC mode.

show processes {*job-id* | *process-name* | **aborts** | **all** | **blocked** | **boot** | **cpu** | **distribution** *process-name* | **dynamic** | **failover** | **family** | **files** | **location** *node-id* | **log** | **mandatory** | **memory** | **pidin** | **searchpath** | **signal** | **startup** | **threadname**} [**location** *node-id*] [**detail**] [**run**]

Syntax Description

job-id	Job identifier for which information for only the process instance associated with the *job-id* argument is displayed.
process-name	Process name for which all simultaneously running instances are displayed, if applicable.
aborts	Displays process abort information.
all	Displays summary process information for all processes.
blocked	Displays details about reply, send, and mutex blocked processes.
boot	Displays process boot information.
cpu	Displays CPU usage for each process.
distribution	Displays the distribution of processes.
dynamic	Displays process data for dynamically created processes.
failover	Displays process failover information.
family	Displays the process session and family information.
files	Displays information about open files and open communication channels.
location *node-id*	Displays information about the active processes from a designated node. The *node-id* argument is entered in the *rack*/*slot*/*module* notation.
log	Displays process log.
mandatory	Displays process data for mandatory processes.
memory	Displays information about the text, data, and stack usage for processes.
pidin	Displays all processes using the QNX command.
searchpath	Displays the search path.
signal	Displays the signal options for blocked, pending, ignored, and queued signals.
startup	Displays process data for processes created at startup.
threadname	Displays thread names.
detail	(Optional) Displays more detail. This option is only available with the *process-name* argument.
run	(Optional) Displays information for only running processes. This option is only available with the *process-name* argument.

Command Default

No default behavior or values

Syntax Description

EXEC
Administration EXEC

Command History

Release	Modification
Release 2.0	This command was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	This command was first supported on the Cisco XR 12000 Series Router. The command was made available in administration EXEC mode.
Release 3.3.0	No modification.
Release 3.4.0	No modification.
Release 3.5.0	The use of this command with no keywords or arguments is not supported.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

Use the **show processes** command to display general information about the active processes. To display more detailed information for a process, specify a job ID or process for the *job-id* argument or *process-name* argument, respectively.

You can also use the **monitor processes** and **monitor threads** commands to determine the top processes and threads based on CPU usage.

Task ID

Task ID	Operations
basic-services	read

Examples

The **show processes** *process-name* command displays detailed information about a process:

```
RP/0/RP0/CPU0:router# show processes ospf

                  Job Id: 261
                     PID: 139453
         Executable path: /crs-rout-0.44.0/bin/ospf
              Instance #: 1
              Version ID: 00.00.0000
                 Respawn: ON
           Respawn count: 1
  Max. spawns per minute: 12
            Last started: Wed Mar 17 07:46:26 2004
           Process state: Run
           Package state: Normal
       Started on config: cfg/gl/ipv4-ospf/proc/100/ord_a/routerid
                    core: TEXT SHAREDMEM MAINMEM
                Max. core: 0
                Mandatory: ON
                Placement: ON
             startup_path: /pkg/startup/ospf.startup
```

```
        Process cpu time: 0.410 user, 0.183 kernel, 0.593 total
JID    TID  LastCPU  Stack pri state        HR:MM:SS:MSEC NAME
261    1    0       40K  10 Receive       0:00:00:0397 ospf
261    2    1       40K  10 Receive       0:00:00:0003 ospf
261    3    0       40K  10 Receive       0:00:00:0007 ospf
261    4    1       40K  10 Condvar       0:00:00:0000 ospf
 --More--
```

Table 81 describes the significant fields shown in the display.

Table 81 show processes Field Descriptions

Field	Description
Job id	Job ID. This field remains constant over process restarts.
PID	Process ID. This field changes when process is restarted.
Executable path	Path for the process executable.
Instance	There may be more than one instance of a process running at a given time (each instance may have more than one thread).
Version ID	API version.
Respawn	ON or OFF. The field indicates if this process restarts automatically in case of failure.
Respawn count	Number of times this process has been (re)started (that is, the first start makes this count 1).
Max. spawns per minute	Number of respawns not to be exceeded in 1 minute. If this number is exceeded, the process stops restarting.
Last started	Date and time the process was last started.
Process state	Current state of the process.
Started on config	Configuration command that started (or would start) this process.
core	Memory segments to include in core file.
Max. core	Number of times to dump a core file. 0 = infinity.

The **show processes memory** command displays details of memory usage for a given process or for all processes, as shown in the following example:

```
RP/0/RP0/CPU0:router# show processes memory

JID    Text     Data     Stack    Dynamic  Process
55     28672    4096     69632    17072128 eth_server
164    143360   4096     20480    13238272 crs_fgid_server
317    167936   4096     45056    10526720 syslogd
122    512000   4096     77824    9797632  bgp
265    57344    4096     57344    5877760  parser_server
254    40960    4096     143360   3084288  netio
63     8192     4096     24576    2314240  nvram
314    4096     4096     36864    1699840  sysdb_svr_local
341    495616   4096     40960    1576960  wdsysmon
259    53248    4096     28672    1490944  nvgen_server
189    32768    4096     32768    1425408  hd_drv
69     77824    4096     110592   1421312  qnet
348    323584   4096     40960    1392640  ospf
347    323584   4096     40960    1392640  ospf
346    323584   4096     40960    1392640  ospf
```

```
345     323584   4096      40960     1392640  ospf
344     323584   4096      40960     1392640  ospf
261     323584   4096      40960     1392640  ospf
 --More--
```

Table 82 describes the significant fields shown in the display.

Table 82 show processes memory Field Descriptions

Field	Description
JID	Job ID.
Text	Size of text region (process executable).
Data	Size of data region (initialized and uninitialized variables).
Stack	Size of process stack.
Dynamic	Size of dynamically allocated memory.
Process	Process name.

The show processes command with the **all** keyword displays summary information for all processes as shown in the following example:

```
RP/0/RP0/CPU0:Router# show processes all

JID    LAST STARTED            STATE    RE-      PLACE-  MANDA-  MAINT- NAME(IID) ARGS
                                        START    MENT    TORY    MODE
-------------------------------------------------------------------------------------
82     03/16/2007 14:54:52.488 Run      1               M       Y      wd-mbi(1)
58     03/16/2007 14:54:52.488 Run      1               M       Y      dllmgr(1)-r 60 -u 30
74     03/16/2007 14:54:52.488 Run      1               M       Y      pkgfs(1)
57     03/16/2007 14:54:52.488 Run      1                       Y      devc-conaux(1) -h -d
librs232.dll -m libconaux.dll -u libst16550.dll
76     03/16/2007 14:54:52.488 Run      1                       Y      devc-pty(1) -n 32
56     Not configured          None     0                       Y      clock_chip(1) -r -b
 --More--
```

Table 83 describes the significant fields shown in the display.

Table 83 show processes all Field Descriptions

Field	Description
JID	Job ID.
Last Started	Date when the process was last started.
State	State of the process.
Restart	Number of times the process has restarted since the node was booted. If a node is reloaded, the restart count for all processes is reset. Normally, this value is 1 because usually processes do not restart. However, if you restart a process using the **process restart** command, the restart count for the process increases by one.
Placement	Indicates whether the process is a placeable process or not. Most processes are not placeable, so the value is blank. ISIS, OSPF, and BGP are examples of placeable processes.

Table 83 show processes all Field Descriptions (continued)

Field	Description
Mandatory	M indicates that the process is mandatory. A mandatory process must be running. If for some reason a mandatory process cannot be started (for example, sysmgr starts it but it keeps crashing), after five attempts the sysmgr causes the node to reload in an attempt to correct the problem. A node cannot function properly if a mandatory process is not running.
Maint Mode	Indicates processes that should be running when a node is in maintenance mode. Maintenance mode is intended to run as few processes as possible to perform diagnostics on a card when a problem is suspected. However, even the diagnostics require some services running.
Name (IID)	Name of the process followed by the instance ID. A process can have multiple instances running, so the IID is the instance ID.
Args	Command-line arguments to the process.

Related Commands

Command	Description
monitor processes	Displays interactive auto-updating process statistics in a full-screen mode.
monitor threads	Displays auto-updating process and thread statistics in a full-screen mode.

slow-migration-interval

To set the slow migration interval for the program, use the **slow-migration-interval** command in placement program configuration mode. To return the slow migration interval to its default value, use the **no** form of this command.

slow-migration-interval *length*

no slow-migration-interval

Syntax Description

length	Length of the slow migration interval in seconds.

Defaults

length: 1 second

Command Modes

Placement program configuration

Command History

Release	Modification
Release 3.3.0	This command was introduced on the Cisco CRS-1.
Release 3.4.0	No modification.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

The **placement reoptimize** command identifies processes that must be moved to achieve optimal placement. To prevent disruption on the router, you can use the **slow-migration-interval** command to specify the time to wait after moving each process, to allow the router to recover before preceding to move the next process.

Task ID

Task ID	Operations
sysmgr	read, write

Examples

The following example shows how to set the slow migration interval:

```
RP/0/RP0/CPU0:router(config)# placement program ipv4_rib
RP/0/RP0/CPU0:router(config-place)# slow-migration-interval 240
```

Related Commands

Command	Description
placement program	Enters placement program configuration mode to set process affinities (preferences).