IBM i
Version 7 Release 4

# *Availability*
# *High availability technologies*

**Note**

Before using this information and the product it supports, read the information in "Notices" on page 35.

# Contents

# High availability technologies

Whether you need high availability for your business applications or are looking to reduce the amount of time it takes to perform daily backups, IBM i high availability technologies provide the infrastructure and tools to help achieve your goals.

All IBM PowerHA® SystemMirror® for i technologies, as well as some Business Partner high availability products, are built on IBM i cluster resource services, or more simply clusters. Cluster provides the underlying infrastructure that allows resilient resources, such as data, devices and applications, to be automatically or manually switched between systems. It provides failure detection and response, so that in the event of an outage, cluster resource services responds accordingly, keeping your data safe and your business operational.

The other key technology that is used by PowerHA technologies is independent disk pools. Independent disk pools allow data and applications be stored on disks that might or might not be present when the system is operational. When independent disk pools are a part of a cluster, the data and applications that are stored within them can be switched to other systems or logical partitions. There are several different technologies that are based on independent disk pools, including switched disks, geographic mirroring, Metro Mirror, Global Mirror, and DS8000® HyperSwap® with independent auxiliary storage pools (IASPs). Determining which technology to base your high availability solution upon depends on several factors. The topic, High availability overview, provides a high-level comparison of these technologies along with criteria that you can use to determine which technologies are most suited for your needs.

In IBM Knowledge Center, the terms IASP and independent disk pool are synonymous.

This topic provides descriptions of key high availability technologies, their associated concepts, and describes the different high availability management interfaces that are supported by IBM i systems.

## What's new for IBM i 7.4

Read about new or changed information for the High Availability (HA) technologies topic collection.

### Copy Services Manager (CSM) functionality enhancements

Support for CSM for DS8000 Metro Mirror and Global Mirror and HyperSwap with Global Mirror sessions.Currently, there is no support for CSM from within the PowerHA GUI.

### HyperSwap functionality enhancements

Improvements and expansion of HyperSwap functionality in PowerHA

- Integrated recovery improvements made to HyperSwap with LUN level switching components.
- HyperSwap functionality with Global Mirror for site resiliency now available via the CSM storage controller support.

Currently, there is no support for HyperSwap with Global Mirror from within the PowerHA GUI.

### High Availability (HA) cluster policies

Introducing HA policies to help PowerHA users modify the cluster behavior in their high availability environment. Policies can be classified by the section of the cluster environment they manage, for example, the Cluster Resource Group (CRG), or communications. Currently, there is no support for HA policies from within the PowerHA GUI.

### Automated management of the Cluster Administrative Domain (CAD)

The administrative domain HA policies, QCST_AD_CREATE, QCST_AD_DELETE, and QCST_AD_RESTORE enable users to:

- automate the addition of Modified Resource Entries (MREs) across nodes in the administrative domain.
- automate deletion of MRE entries across the nodes in the administrative domain.
- restore MREs in the administrative domain, if required.

### Additional cluster management features

Two additional HA policies assist users with cluster management:

- The CRG HA policy, QCST_CRG_CANCEL_FAILOVER integrates control over the ability to cancel automatic CRG failovers.
- The communication policy, QHA_COMM_STRICT_CERT_CHECK allows users to change the default security settings between the cluster nodes and storage devices.

### How to see what's new or changed

To help you see where technical changes have been made, this information uses:

- The » image to mark where new or changed information begins.
- The « image to mark where new or changed information ends.

In PDF files, you might see revision bars (|) in the left margin of new and changed information.

To find other information about what's new or changed in this release, see the Memo to users.

## PDF file for High availability technologies

You can view and print a PDF file of this information.

To view or download the PDF version of this document, select High availability technologies

You can view or download these related topic PDFs:

- High availability overview

  contains the following topics:
  - Benefits of high availability
  - High availability criteria
  - Components of high availability
  - IBM PowerHA SystemMirror for i overview
- Implementing high availability

  contains the following topics:
  - Installing IBM PowerHA SystemMirror for i (iHASM) licensed program (5770-HAS)
  - Uninstalling IBM PowerHA SystemMirror for i (iHASM) licensed program (5770-HAS)
  - Planning your high availability solution
  - Implementing PowerHA
  - Managing PowerHA
  - Troubleshooting your high availability solution

### Saving PDF files

To save a PDF on your workstation for viewing or printing:

1. Right-click the PDF link in your browser.
2. Click the option that saves the PDF locally.

3. Navigate to the directory in which you want to save the PDF.
4. Click **Save**.

### Downloading Adobe Reader

You need Adobe Reader installed on your system to view or print these PDFs. You can download a free copy from the Adobe Web site (www.adobe.com/products/acrobat/readstep.html).

# IBM i Cluster technology

As companies strive to compete in today's environment, high availability has become essential to many businesses. The IBM i cluster technology can be used to achieve high availability in IBM i environments. Cluster technology provides mechanisms that enable critical resources to be automatically available on backup systems. Those resources could include data, application programs, devices, or environment attributes.

In order for those business applications to be highly available, multiple systems might be required. This type of distributed computing environment can become complex to manage. A cluster can simplify this complexity. In IBM i, a cluster is a set or collection of systems or logical partitions, called cluster nodes. A cluster provides a means to monitor and manage the environment that provides high availability for the business application. A cluster can be a simple, two-node high availability environment for a specific business application or it can be a more complex, multiple system environment for multiple disjoint applications. A cluster might consist of many nodes, while a specific application might only be dependent on a subset of those nodes. An application on a node could fail without the entire node failing. Cluster technology provides the mechanisms for defining resilient resources in an environment, detecting outages, and responding to these failures. It provides the critical infrastructure that enables high availability solutions.

**Related information**

Planning clusters

Configuring clusters

Managing clusters

## Cluster concepts

An IBM i *cluster* is a collection of one or more systems or logical partitions that work together as a single system. Use this information to understand the elements and their relationship to each other.

### Cluster node

A *cluster node* is a IBM i system or logical partition that is a member of a cluster.

When you create a cluster, you specify the systems or logical partitions that you want to include in the cluster as nodes. Each cluster node is identified by a 1 to 8 character cluster node name which is associated with one or two IP addresses that represent the system. When configuring a cluster, you can use any name that you want for a node in the cluster. However, it is recommended that the node name be the same as the host name or the system name.

Cluster communication makes use of the TCP/IP protocol suite to provide the communications paths between cluster services on each node in the cluster. The set of cluster nodes that are configured as part of the cluster is referred to as the cluster membership list.

**Related information**

Configuring nodes

Managing nodes

### Cluster resource group (CRG)

A *cluster resource group (CRG)* is an IBM i system object that is a set or grouping of cluster resources that are used to manage events that occur in a high availability environment.

A cluster resource is a resource that is required to be highly available for the business. Cluster resources can be either moved or replicated to one or more nodes within a cluster. Examples are a payroll application, data library, or disk units. A collection of cluster resources can be monitored and managed by a CRG. A CRG also defines the relationship between nodes that are associated with the cluster resources. This management includes specifying which nodes the resources can be on, which node currently has the resources, and which node should get the resources if a failure should occur.

The IBM i cluster defines four types of CRGs: device, application, data, and peer. Each type is designed to monitor and manage a specific type of cluster resource. As an example, a business application may typically have two cluster resources, an application and its associated data. An application CRG could be used to manage the application resource. The data could either be managed by a device CRG if the data is stored in switched disks, or a data CRG which could use a high availability business partner application to replicate the data between nodes.

Within these types of CRGs, there are two common elements: a recovery domain and an exit program. The CRG manages resource availability across a subset of nodes within the cluster, called a recovery domain.

The exit program performs actions when the CRG detects certain events, such as a new node being added to the recovery domain, or the current primary node failing.

**Related information**

Configuring CRGs

Managing cluster resource groups (CRGs)

#### *Application CRG*

In IBM i high availability environments, application resiliency, which is the ability to restart an application on a backup system, is supported through the application cluster resource group (CRG). The takeover IP address allows access to the application without regard on which system the application is currently running. This capability allows resilient applications to be switched from one node to another in the event of an outage.

An application CRG can start an application as well as monitor for application failure. An application is defined to be a program or set of programs that can be called to deliver some business solution. The application CRG does not manage any data associated with the application. The data would be managed by a data or device CRG. The exit program in an application CRG serves two purposes. First, the exit program is invoked when cluster events occur to allow processing that is specific to that application for these events. The second purpose of the exit program is to start and then monitor the health of the actual application program. Although some business applications are created internally, others are purchased from outside vendors. If an application provider has made their application highly available, they would provide the CRG exit program along with the application. The exit program would be written to react appropriately to cluster events and manage the application.

**Related information**

Planning application resiliency

Creating application CRGs

#### *Data CRG*

A data cluster resource group (CRG) is an IBM i system object which assists in data replication between the primary and backup nodes in the recovery domain. A data CRG does not do the replication, but uses the exit program to inform a replication program when to start or end replication, and on which nodes to replicate. A data CRG does not monitor for a data resource failure.

Data CRGs are primarily used with logical replication applications, which are provided by several high availability IBM Business Partners.

**Related information**

Creating data CRGs

### *Device CRG*

A device cluster resource group (CRG) supports device resiliency in IBM i high availability environments. A device CRG contains one or a list of independent auxiliary storage pools (IASPs) which are replicated or switched between systems. IASPs can only be switched or replicated between nodes in the recovery domain for the device CRG. Access to the entire list of IASPs in the device CRG is switched to the backup node when an outage, planned or unplanned, occurs. When creating the environment, you can either create a new IASP or use an existing one.

There are IBM PowerHA SystemMirror for i technologies available to switch or replicate the data in the IASPs. The switchable logical unit (LUN) technology switches access of one copy of the IASP from the primary to the backup server. With replication technologies such as geographic mirroring, Metro Mirror, or Global Mirror, data is mirrored from the IASP at the production site to another IASP at the backup site. Usually these sites are geographically separate from one another, providing disaster recovery protection. In these environments, the device CRG controls switching access between mirrored copies of the IASP. When the production site experiences an outage event, the device CRG switches access to the IASP to the backup site.

**Related information**

Creating device CRGs

### *Peer CRG*

A peer cluster resource group (CRG) is a non-switchable cluster resource group in which each IBM i node in the recovery domain plays an equal role in the recovery of the node. The peer cluster resource group provides peer resiliency for groups of objects or services.

Unlike the other CRG types, in which only the primary CRG node is the node doing the work, in a peer CRG all of the nodes in the recovery domain work together. The purpose of peer CRG is to provide a general distributed-computing framework for which programmers can write applications. The peer cluster resource group provides peer resiliency for groups of objects or services. The end user, end-user applications, and business partner applications, not the system, choose the groups of objects.

**Related information**

Creating peer CRGs

### *Recovery domain*

Within IBM i clusters technology, a *recovery domain* is a subset of cluster nodes that are grouped together in a cluster resource group (CRG) for a common purpose such as performing a recovery action or synchronizing events.

There are two basic recovery domain models that can be used in high availability environments. These models are based on the type of cluster resource group that is created and the roles that are defined in the recovery domain. With the primary-backup model, users must define the node as either a primary, backup, or replicate role. Device, application and data CRGs support these role definitions. These roles are defined and managed within the recovery domain.

If a node has been defined as the primary access point for the resource, then other nodes provide backup if the primary node fails. Nodes defined as backups are nodes capable of being the access point for the resource. There is a specified order of backup nodes, which determines which backup would be first in line to be the primary should the existing primary fail. For primary-backup models, IBM i clusters will automatically respond when a node fails or switches over, based on these role definitions. For example, if Node A, which is designated as the primary, fails, Node B, which is defined as the first backup, becomes the new primary. Other nodes defined as backups will be reordered accordingly.

A replicate node is similar to a backup node but is not capable of being an access point for a resource (i.e. can not become a primary). The most common use of a replicate node is in a data CRG, where the data could be made available on a replicate node for report generation, although that node would never become the primary node.

The second recovery domain model is peer. With peer model, there is no ordered recovery domain. For a peer model, nodes can be defined as either peer or replicate. Peer CRGs support these role definitions. If nodes are defined as peer, then all the nodes in the recovery domain are equal and can provide the access point for the resource. However, there is no specified order during an outage of a peer node. The recovery domain nodes are notified when other nodes fail or have outages, but since there is no automatic response to these events, it is necessary for an application to provide actions for those events.

The four types of roles a node can have in a recovery domain are:

**Primary**

The cluster node that is the primary point of access for the cluster resource.

- For a data CRG, the primary node contains the principle copy of a resource.
- For an application CRG, the primary node is the system on which the application is currently running.
- For a device CRG, the primary node is the current owner of the IASP.
- For a peer CRG, the primary node is not supported.

If the primary node for a CRG fails, or a manual switchover is initiated, then the primary point of access for that CRG is moved to the first backup node.

**Backup**

The cluster node that will take over the role of primary access if the present primary node fails or a manual switchover is initiated.

- For a data CRG, this cluster node contains a copy of that resource which is kept current with replication.
- For a peer CRG, the backup node is not supported.

**Replicate**

A cluster node that has copies of cluster resources, but is unable to assume the role of primary or backup. Failover or switchover to a replicate node is not allowed. If you ever want a replicate node to become a primary, you must first change the role of the replicate node to that of a backup node.

- For peer CRGs, nodes defined as replicate represent the inactive access point for cluster resources.

**Peer**

A cluster node which is not ordered and can be an active access point for cluster resources. When the CRG is started, all the nodes defined as peer will be an active access point.

- For a peer CRG, the access point is controlled entirely by the management application and not the system. The peer role is only supported by the peer CRG.

### *Cluster resource group exit programs*

In IBM i high availability environments, *cluster resource group exit programs* are called after a cluster-related event for a CRG occurs and responds to the event.

An exit program is called when a CRG detects certain events, such as a new node being added to the recovery domain, or the current primary node failing. The exit program is called with an action code that indicates what the event is. Furthermore, the exit program has the capability to indicate whether to process the event or not. User-defined simply means the IBM i cluster technology does not provide the exit program. Typically the exit program is provided by the application or data replication provider. The exit program is the way a CRG communicates cluster events to the exit program provider. The exit program can perform the appropriate action based on the event, such as allowing a resource access point to move to another node. The exit program is optional for a resilient device CRG but is required for the other CRG types. When a cluster resource group exit program is used, it is called on the occurrence of cluster-wide events, including when:

- A node leaves the cluster unexpectedly.
- A node leaves the cluster as a result of the `End Cluster Node (QcstEndClusterNode)` API or `Remove Cluster Node Entry (QcstRemoveClusterNodeEntry)` API.
- The cluster is deleted as a result of the `Delete Cluster (QcstDeleteCluster)` API.
- A node is activated by the `Start Cluster Node (QcstStartClusterNode)` API.
- Communication with a partitioned node is re-established.

Exit programs are written or provided by cluster middleware IBM Business Partners and by cluster-aware application program providers.

For detailed information on the cluster resource group exit programs, including what information is passed to them for each action code, see Cluster Resource Group Exit Program in the cluster API documentation.

### Cluster version

A *cluster version* represents the level of function available on the cluster.

Versioning is a technique that allows the cluster to contain nodes at multiple release levels and fully interoperate by determining the communications protocol level to be used.

**Note:** If you are using the IBM PowerHA SystemMirror for i licensed program number (5770-HAS), cluster version level 6 or higher is required.

There are actually two cluster versions:

**Potential cluster version**

Represents the most advanced level of cluster function available for a given node. This is the version at which the node is capable of communicating with the other cluster nodes.

**Current cluster version**

Represents the version currently being used for all cluster operations. This is the version of communications between the nodes in the cluster.

The potential cluster version is incremented on every IBM i release which has significant new clustering functionality not available in earlier cluster versions. If the current cluster version is less than the potential cluster version, then that function cannot be used since some nodes cannot recognize or process the request. To take advantage of such new function, every node in the cluster needs to be at the same potential cluster version and the current cluster version must also be set to that level.

When a node attempts to join a cluster, its potential cluster version is compared against the current cluster version. If the value of the potential cluster version is not the same as current (N) or not equal to the next version level (N+2), then the node is not allowed to join the cluster. Note that the current cluster version is initially set by the first node defined in the cluster using the value specified on the create cluster API or command.

For example if you want 7.2 nodes to exist with 7.3 nodes you can do one of the following:

- Create the cluster on a 7.2 node and add in the 7.3 node.
- Create the cluster on a 7.3 node specifying to allow previous nodes to be added to the cluster, then add 7.2 nodes to the cluster.

In a multiple-release cluster, cluster protocols will always be run at the lowest node release level, the current cluster version. This is defined when the cluster is first created. N can either be set to the potential node version running on the node that originated the create cluster request or one cluster version previous to the originators potential node version. Nodes in the cluster can differ by at most two cluster version level.

Once all nodes in the cluster have been upgraded to the next release, the cluster version can be upgraded so that new functions are available. This can be accomplished by adjusting the cluster version.

**Attention:** If the new version of the cluster is not equal, one or two versions higher to the current cluster version, then the cluster node will fail when it is restarted. To recover from this situation, the cluster on that node must be deleted and the cluster version adjusted before the node can be re-added to the cluster.

**Attention:** When you are using independent auxiliary storage pools (IASPs) in your cluster, there are restrictions in doing a switchover between releases. If an independent disk pool is switched from a node running a previous release of IBM i to a node running the current release of IBM i, when it is made available on the node, its internal contents are changed and it cannot be made available to the previous release node again.

Read more on cluster versions in the Clusters APIs documentation, including information about restrictions and how cluster versions correspond to IBM i releases.

**Related information**

Planning mixed-release clusters

Adjusting the cluster version

Scenario: Upgrading operating system in an high availability environment

### Device domain

A *device domain* is a subset of nodes in an IBM i cluster that share device resources. More specifically, nodes in a device domain can participate in a switching action for some collection of resilient device resources.

Device domains are identified and managed through a set of interfaces that allow you to add a node to a device domain or remove a node from a device domain.

Device domains are used to manage certain global information necessary to switch a resilient device from one node to another. All nodes in the device domain need this information to ensure that no conflicts occur when devices are switched. For example, for a collection of switched disks, the independent disk pool identification, disk unit assignments, and virtual address assignments must be unique across the entire device domain.

A cluster node can belong to only one device domain. Before a node can be added to the recovery domain for a device CRG, the node must be first defined as a member of a device domain. All nodes that will be in the recovery domain for a device CRG must be in the same device domain.

To create and manage device domains, you must have Option 41 (IBM i - HA Switchable Resources) installed and a valid license key on your system.

**Related information**

Adding a node to a device domain

### Cluster jobs

When managing an IBM i cluster, you need to know about cluster job structures and how they are organized on the system.

#### Cluster resource services jobs

Cluster resource services consists of a set of multi-threaded jobs. Critical cluster resource services jobs are system jobs and run under the QSYS user profile. Several work management-related functions, such as ending a job (ENDJOB), are not allowed on system jobs. This mean that a users cannot inadvertently end one of these cluster system jobs, causing problems in the cluster and high availability environment. When clustering is active on a system, the following jobs run as system jobs:

- Cluster control job consists of one job that is named QCSTCTL.
- Cluster resource group manager consists of one job that is named QCSTCRGM.

  **Note:** The QCSTCTL and QCSTCRGM job are cluster critical jobs. That is, the jobs must be running in order for the node to be active in the cluster.
- Each cluster resource group consists of one job per cluster resource group object. The job name is the same as the cluster resource group name.
- Cluster administrative domain jobs consist of a single system job running on every node in the cluster. The name of the system job is the name of the cluster administrative domain.

It is important to note that some work management actions will end these cluster system jobs, causing a failover to occur. During these actions, clustering ends and failover occurs, based on how that node is defined in the CRG. See the topic, Example: Failover outage events, for a complete list of system-related events that cause failovers.

You can use the Change Cluster Recovery (CHGCLURCY) command to restart the cluster resource group job that ended without ending and restarting clustering on a node.

Several other less critical cluster-related jobs are part of the QSYSWRK subsystem. Ending this QSYSWRK subsystem, ends these jobs without causing failover, however they can cause cluster problems, which may require a recovery action. Some of these jobs run under the QSYS user profile.

Most cluster resource group APIs result in a separate job being submitted that uses the user profile specified when the API was invoked. The exit program defined in the cluster resource group is called in the submitted job. By default, the jobs are submitted to the QBATCH job queue. Generally, this job queue is used for production batch jobs and will delay or prevent completion of the exit programs. To allow the APIs to run effectively, create a separate user profile, job description, and job queue for use by cluster resource groups. Specify the new user profile for all cluster resource groups that you create. The same program is processed on all nodes within the recovery domain that is defined for the cluster resource group.

A separate batch job is also submitted for a cluster administrative domain when a cluster resource group API is called. The IBM supplied QCSTADEXTP program is called. The submitted job runs under the QCLUSTER user profile using the QCSTJOBD job description.

**Related information**

Example: Managing failover outage events

Cluster APIs Use of User Queues

System jobs

### Container cluster resource group

Manage multiple Cluster Resource Groups (CRGs) with the CRG container

#### About CRG containers

A CRG container serves as the control object for a collection of CRGs managed as a single entity for HA operations.

CRG containers can manage device, data, and application CRGs. Similar to the CRGs it manages, the CRG container provides the ability to perform a switchover or failover locally or across sites. A CRG managed by a CRG container is considered a managed CRG. Managed CRGs must maintain consistency with the managing CRG container.

#### Configuration requirements

For a CRG container to manage CRGs, and for CRGs to remain consistent with the CRG container, these requirements must be in place:

**Container recovery domain**
Within the CRG container, all managed CRG recovery domains must be a subset of the CRG container recovery domain. A CRG container's recovery domain must have all the nodes located in every managed CRGs' recovery domain.

**Cluster nodes**
The recovery domain of a CRG managed by a container must have all of its nodes located in the CRG container recovery domain.

**CRG**
Only device, data, and application CRGs can be managed by a CRG container. A managed CRG can be in only one container.

**Site**
In the HA environment, a site contains a subset of recovery domain nodes in the same physical location. The CRG parameter SITE is required when creating a CRG container. The CRG container is designed to perform its functions at a site level and require nodes to be associated with named sites in their recovery domains, representing the nodes in the same physical location. Managed CRGs using sites in their recovery domains must match the site names that the container CRG uses. For CRG types Data and Application that do not contain sites, the CRG container logically divides these into sites. For example, a container Site1 has nodes A and B, and Site2 has nodes C and D. A data CRG has nodes A

and C, which the container would manage as though the data CRG had a Site1 of node A and a Site2 of node C.

**Consistency**

A managed CRG is considered consistent with its container CRG only if the managed CRG has the same source site as the container.

These rules ensure that the CRG container maintains the management functions for the CRGs within it.

### CRG container status

A CGR container can have an active, inactive, or indoubt status depending on the statuses of the CRGs it manages. Like CRGs, a CRG container must be active to provide protection and switching capabilities.

### CRG container commands

CRG containers rely on much of the underlying technology that CRGs use and thus utilize many of the same commands that the CRGs use. Many functions of the CRG container can be managed from the Work with Cluster Resource Groups screen used to create and manage CRGs. Some commands however, are different for containers even though they may perform a task similar to a CRG command. CRG containers use a CNR suffix to distinguish the command is used on a CRG container. For example, the Start CRG (STRCRG) command for a CRG container is STRCRGCNR, Start CRG container. Table below lists related commands between CRGs and CRG containers.

| CRG command | CRG container command |
|---|---|
| Create CRG (CRTCRG) | Create CRG container (CRTCRGCNR) |
| Delete CRG (DLTCRG) | Delete CRG container (DLTCRGCNR) |
| Start CRG (STRCRG) | Start CRG container (STRCRGCNR) |
| End CRG (ENDCRG) | End CRG container (ENDCRGCNR) |
| Display CRG information (DSPCRGINF) | Display CRG container (DSPCRGCNR) |
|  | Configure CRG container (CFGCRGCNR) |
| Change CRG primary (CHGCRGPRI) | Change CRG primary (CHGCRGPRI) |

To learn more about CRG containers and how to work with them, see Creating and configuring a Cluster Resource Group container in the Implementing High Availability section of the Knowledge Center.

## Base cluster functions

Several basic IBM i cluster functions monitor the systems within the cluster to detect and respond to potential outages in the high availability environment.

Cluster resource services provide a set of integrated services that maintain cluster topology, perform heartbeat monitoring, and allow creation and administration of cluster configuration and cluster resource groups. Cluster resource services also provides reliable messaging functions that keep track of each node in the cluster and ensure that all nodes have consistent information about the state of cluster resources.

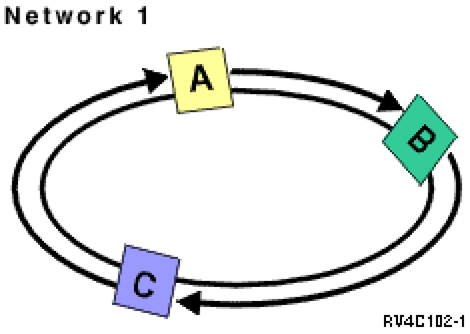
### Heartbeat monitoring

*Heartbeat monitoring* is an IBM i cluster base function that ensures that each node is active by sending a signal from every node in the cluster to every other node in the cluster to convey that they are still active.

When the heartbeat for a node fails, cluster resource services takes the appropriate action.

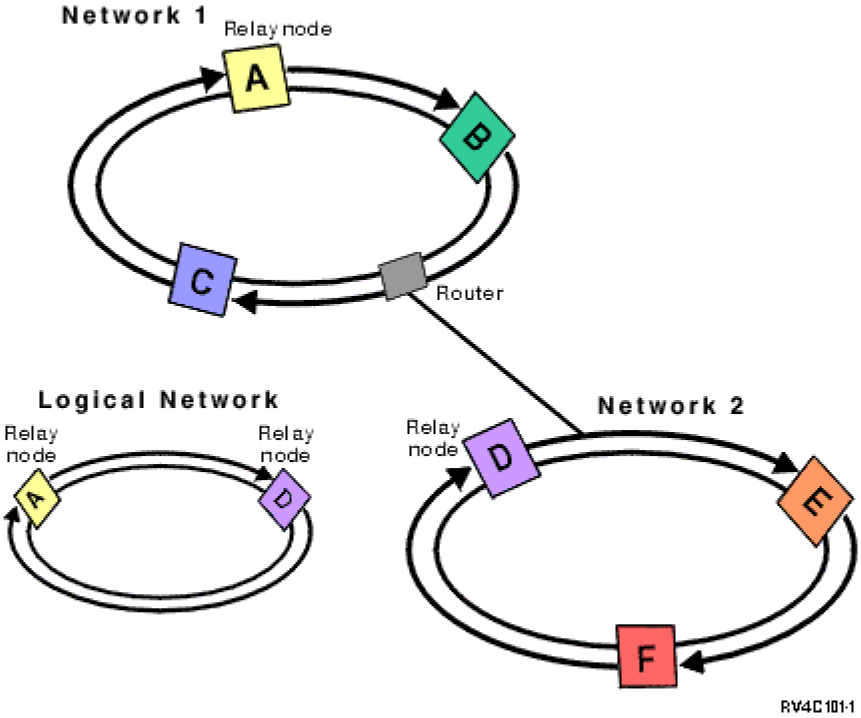
Consider the following examples to understand how heartbeat monitoring works:

**Example 1**



With the default (or normal) settings, a heartbeat message is sent every 3 seconds from every node in the cluster to its upstream neighbor. For example, if you configure Node A, Node B, and Node C on Network 1, Node A sends a message to Node B, Node B sends a message to Node C, and Node C sends a message to Node A. Node A expects an acknowledgment to the heartbeat from Node B as well as an incoming heartbeat from the downstream Node C. In effect, the heartbeating ring goes both ways. If Node A did not receive a heartbeat from Node C, Node A and Node B continue to send a heartbeat every 3 seconds. If Node C missed four consecutive heartbeats, a heartbeat failure is signaled.

**Example 2**



Let's add another network to this example to show how routers and relay nodes are used. You configure Node D, Node E, and Node F on Network 2. Network 2 is connected to Network 1 using a router. A router can be another IBM i machine or a router box that directs communications to another router somewhere else. Every local network is assigned a relay node. This relay node is assigned to the node that has the lowest node ID in the network. Node A is assigned as the relay node on Network 1, and Node D is assigned as the relay node on Network 2. A logical network containing Node A and Node D is then created. By using routers and relay nodes, the nodes on these two networks can monitor each other and signal any node failures.

### Reliable message function

The *reliable message function* of cluster resource services keeps track of each node in an IBM i cluster and ensures that all nodes have consistent information about the state of cluster resources.

Reliable messaging uses retry and timeout values that are unique to clustering. These values are preset to values that should suit most environments. However, they can be changed through the Change cluster resource services settings interface. The message retry and timeout values are used to determine how many times a message is sent to a node before a failure or partition situation is signaled. For a local area network (LAN), the amount of time it takes to go through the number of retries before a failure or partition condition is signaled is approximately 45 seconds using the default retry and timeout values. For a remote network, more time is allowed to determine whether a failure or partition condition exists. You can figure approximately 4 minutes and 15 seconds for a remote network.

## Cluster events

Cluster events are actions and events that can happen in an IBM i high availability environment and to which cluster resource services responds.

Cluster resource services provides detection and response for particular events in a high availability environment.

### Switchover

*Switchover* happens when you manually switch access to a resource from one IBM i system to another.

You usually initiate a manual switchover if you wanted to perform system maintenance, such as applying program temporary fixes (PTFs), installing a new release, or upgrading your system. Contrast this with a failover, which happens automatically when a outage occurs on the primary node.

When a switchover occurs, access is switched from the cluster node currently acting as the primary node in the recovery domain of the cluster resource group to the cluster node designated as the first backup. See recovery domain for information on how the switchover order is determined.

If you are doing an administrative switchover of multiple CRGs, the order you specify should consider the relationships between the CRGs. For example, if you have an application CRG that depends on data associated with a device CRG, the steps to an ordered switchover are:

1. Quiesce the application on the old primary node (to quiesce changes to the data).
2. Switch the device CRG to the new primary node.
3. Switch the application CRG to the new primary node.

### Failover

*Failover* occurs when an IBM i system in a cluster automatically switches over to one or more backup nodes in the event of a system failure.

Contrast this with a switchover, which happens when you manually switch access from one server to another. A switchover and a failover function identically once they have been triggered. The only difference is how the event is triggered.

When a failover occurs, access is switched from the cluster node currently acting as the primary node in the recovery domain of the cluster resource group to the cluster node designated as the first backup.

When multiple cluster resource groups (CRGs) are involved in a failover action, the system processes the device CRGs first, the data CRGs second, and the application CRGs last.

With device CRGs, failover processing varies off the devices associated with the CRG. The devices are varied off even if failover is cancelled via the cluster message queue or failover message queue. Some system actions which cause a failover, such as ending TCP/IP, do not affect the entire system, so user and jobs may still need access to the device. You may want to end the CRG before taking those system actions and keep the devices varied on for the following reasons:

- When you are performing a save with Option 21 after ending all subsystems (ENDSBS *ALL).
- When you are performing routine fixes by ending subsystems or ending TCP/IP and do not to spend extra time varying off and on devices.

- When the entire system is not ending, it is possible that other jobs would still need access to the device.

The failover message queue receives messages regarding failover activity for each CRG defined in the cluster. You can also use the cluster message queue to receive a single message for all the CRGs failing over to the same node. Both allow you to control the failover processing of cluster resource groups and nodes. If you have both the cluster message queue and the failover message queue configured, the cluster message queue takes priority. If you prefer failover messages for each CRGs within a cluster, you should not configure the cluster message queue. For either message queue, you can use IBM i watch support to monitor these message queues for activity.

**Related concepts**

Recovery domain
Within IBM i clusters technology, a *recovery domain* is a subset of cluster nodes that are grouped together in a cluster resource group (CRG) for a common purpose such as performing a recovery action or synchronizing events.

### *Cluster message queue*

Within IBM i high availability environments, you can specify a cluster message queue where you can receive and respond to messages that provide details on failover events in the cluster. This message provides information on all cluster resource groups (CRGs), which are failing over to the same node when the primary node for the CRGs ends or fails.

It is similar to failover message queue, but instead of receiving one message per CRG, you receive only one regarding all the CRGs failing over to the same node. If you have both the cluster message queue and the failover message queue configured, the cluster message queue takes priority. If you prefer failover messages for each CRG within a cluster, you should not configure the cluster message queue. For either message queue, you can use IBM i watch support to monitor these message queues for activity.

The following table describes the action that is performed for each of these message queues:

*Table 1. Cluster and failover message queues actions*

| Cluster message queue defined | Failover message queue defined | Response |
|---|---|---|
| No | No | Failover continues with no user action |
| No | Yes | Message (CPABB01) sent to each CRG's failover message queue on the first backup node |
| Yes | No | For node-level failovers, one message (CPABB02) is sent to cluster message queue on the first backup node which controls all CRGs failing over to that node. For crg-level failovers, one message (CPABB01) per CRG will be sent to the cluster message queue on the first backup node which controls the individual CRG failing over to that node. |

*Table 1. Cluster and failover message queues actions (continued)*

| Cluster message queue defined | Failover message queue defined | Response |
|---|---|---|
| Yes | Yes | CRG failover message queue will be ignored. For node-level failovers, one message (CPABB02) is sent to cluster message queue on the first backup node which controls all CRGs failing over to that node. For crg-level failovers, one message (CPABB01) per CRG is sent to the cluster message queue on the first backup node which controls the individual CRG failing over to that node. |

You can define a cluster message queue by providing a name for the queue and a library in which the queue resides. You also can specify the number of minutes to wait for a reply to the failover message. Once this time is exceeded, the specified default failover action will be taken.

**Related concepts**

Failover message queue
The failover message queue receives messages regarding failover activity for CRGs within an IBM i cluster.

**Related information**

Start Watch (STRWCH)

### *Failover message queue*

The failover message queue receives messages regarding failover activity for CRGs within an IBM i cluster.

Using the failover message queue allows an administrator to be notified before a failover occurs. This gives the administrator the ability to cancel the failover if the desired behavior is to prevent the failover at this time. The failover message queue provides messages for each CRG defined within a cluster. You can use IBM i watch support to monitor the failover message queue.

The failover message queue is defined when creating a cluster resource group using the Cluster Resource Service graphical interface in IBM Navigator for i. You can also specify a failover message queue with the **`CRTCRG (Create Cluster Resource Group)`** command and the **`CHGCRG (Change Cluster Resource Group)`** command.

**Note:** To use the Cluster Resource Service graphical interface or the CL commands, you must have IBM PowerHA SystemMirror for i licensed program installed.

It can also be modified using native IBM i cluster resource group APIs. Further details on these APIs can be found in the Cluster Resource Group API information.

**Related concepts**

Cluster message queue
Within IBM i high availability environments, you can specify a cluster message queue where you can receive and respond to messages that provide details on failover events in the cluster. This message provides information on all cluster resource groups (CRGs), which are failing over to the same node when the primary node for the CRGs ends or fails.

**Related information**

Create Cluster Resource Group (CRTCRG) command

Change Cluster Resource Group (CHGCRG) command

Cluster Resource Group APIs

### Cluster partition

Within IBM i high availability environments, a *cluster partition* is a subset of the active cluster nodes which results from a communications failure. Members of a partition maintain connectivity with each other.

A cluster partition occurs in a cluster whenever communication is lost between one or more nodes in the cluster and a failure of the lost nodes cannot be confirmed. When a cluster partition condition is detected, cluster resource services limits the types of actions that you can perform on the nodes in the cluster partition. Restricting function during a partition is done so cluster resource services will be able to merge the partitions once the problem that caused it has been fixed.

Certain CRG operations are restricted when a cluster is partitioned. For details on which operations are restricted for each type of partition, see Cluster Resource Group APIs.

If a cluster administrative domain is partitioned, changes will continue to be synchronized among the active nodes in each partition. When the nodes are merged back together again, the cluster administrative domain will propagate all changes made in every partition so that the resources are consistent within the active domain. You can also specify the rejoin behavior for cluster administrative domain.

### Merge

A *merge* operation is similar to a rejoin operation except that it occurs when nodes that are partitioned begin communicating again.

The partition may be a true partition in that cluster resource services is still active on all nodes. However, some nodes cannot communicate with other nodes due to a communication line failure. Or, the problem may be that a node actually failed, but was not detected as a failure.

In the first case, the partitions are merged back together automatically once the communication problem is fixed. This happens when both partitions periodically try to communicate with the partitioned nodes and eventually reestablish contact with each other. In the second case, you need to change the status of the partitioned node to failed from the active node. You can then restart cluster resource services on that node from any node within the cluster.

#### ***Example: Merge***

Within IBM i cluster technology, a merge operation occurs in several different situations.

A merge operation can occur with one of the following configurations:

*Table 2. Merge between a primary and secondary partition*

| **merge** | |
|---|---|
| primary partition | secondary partition |

*Table 3. Merge between a secondary and secondary partition*

| **merge** | |
|---|---|
| secondary partition | secondary partition |

Primary and secondary partitions are unique to cluster resource groups (CRG). For a primary-backup CRG, a primary partition is defined as the partition that contains the node identified as the primary access point. A secondary partition is defined as a partition that does not contain the node identified as the primary access point.

For peer CRG, if the recovery domain nodes are fully contained within one partition, it will be the primary partition. If the recovery domain nodes span a partition, there will be no primary partition. Both partitions will be secondary partitions.

See Synchronization of monitored resource for information about the behavior of a cluster administrative domain during a rejoin.

*Table 4. Merge between a primary and secondary partition*

| merge operation | | | |
|---|---|---|---|
| primary partition | | secondary partition | |
| contains copy of CRG | does NOT contain copy of CRG | contains copy of CRG | does NOT contain copy of CRG |
| (1) | (2) | (3) | (4) |

During a primary secondary merge as shown in the diagram above, the following situations are possible:

1. 1 and 3
2. 1 and 4
3. 2 and 3 (Cannot happen since a majority partition has the primary node active and must have a copy of the CRG.)
4. 2 and 4 (Cannot happen since a majority partition has the primary node active and must have a copy of the CRG.)

### Primary-secondary merge situations

A copy of the CRG object is sent to all nodes in the secondary partition. The following actions can result on the nodes in the secondary partition:

- No action since the secondary node is not in the CRG's recovery domain.
- A secondary node's copy of the CRG is updated with the data from the primary partition.
- The CRG object is deleted from a secondary node since the secondary node is no longer in the CRG's recovery domain.
- The CRG object is created on the secondary node since the object does not exist. However, the node is in the recovery domain of the CRG copy that is sent from the primary partition.

*Table 5. Secondary and secondary merge scenario*

| merge operation | | | |
|---|---|---|---|
| secondary partition | | secondary partition | |
| contains copy of CRG | does NOT contain copy of CRG | contains copy of CRG | does NOT contain copy of CRG |
| (1) | (2) | (3) | (4) |

During a secondary-secondary merge as shown in the diagram above, the following situations are possible:

1. 1 and 3
2. 1 and 4
3. 2 and 3
4. 2 and 4

### Secondary-secondary merge situation 1

For a primary-backup CRG, the node with the most recent change to the CRG is selected to send a copy of the CRG object to all nodes in the other partition. If multiple nodes are selected because they all appear to have the most recent change, the recovery domain order is used to select the node.

When merging two secondary partitions for peer CRG, the version of the peer CRG with Active status will copied to other nodes in the other partition. If both partitions have the same status for peer CRG, the partition which contains the first node listed in the cluster resource group recovery domain will be copied to nodes in another partition.

The following actions can occur on the receiving partition nodes in either a primary-backup CRG or a peer CRG:

- No action since the node is not the CRG's recovery domain.
- The CRG is created on the node since the node is in the recovery domain of the copy of the CRG object it receives.
- The CRG is deleted from the node since the node is not in the recovery domain of the copy of the CRG object it receives.

#### Secondary-secondary merge situations 2 and 3

A node from the partition which has a copy of the CRG object is selected to send the object data to all nodes in the other partition. The CRG object may be created on nodes in the receiving partition if the node is in the CRG's recovery domain.

#### Secondary-secondary merge situation 4

Internal data is exchanged to ensure consistency throughout the cluster.

A primary partition can subsequently be partitioned into a primary and secondary partition. If the primary node fails, Cluster Resource Service (CRS) detects it as a node failure. The primary partition becomes a secondary partition. The same result occurs if you ended the primary node that uses the End Cluster Node API. A secondary partition can become a primary partition if the primary node becomes active in the partition either through a rejoin or merge operation.

For a merge operation, the exit program is called on all nodes in the CRG's recovery domain regardless of which partition the node is in. The same action code as rejoin is used. No roles are changed as a result of the merge, but the membership status of the nodes in the CRG's recovery domain is changed from *partition* to *active*. Once all partitions merge together, the partition condition is cleared, and all CRG API's can be used.

### Rejoin

*Rejoin* means to become an active member of an IBM i cluster after having been a nonparticipating member.

For example, when clustering is restarted on a node after the node has been inactive, the cluster node rejoins the cluster. You start cluster resource services on a node by starting it from a node that is already active in the cluster. Beginning with cluster version 3, a node can start itself and will be able to rejoin the current active cluster, provided it can find an active node in the cluster. See Start a cluster node for details.

Suppose that nodes A, B, and C make up a cluster. Node A fails. The active cluster is now nodes B and C. Once the failed node is operational again, it can rejoin the cluster when the node is started from any cluster node, including itself. The rejoin operation is done on a cluster resource group basis, which means that each cluster resource group (CRG) joins the cluster independently.

The primary function of rejoin ensures that the CRG object is replicated on all active recovery domain nodes. The rejoining node, as well as all existing active cluster nodes, must have an identical copy of the CRG object. In addition, they must have an identical copy of some internal data.

When a node fails, the continued calling of cluster resource services on the remaining nodes in the cluster can change the data in a CRG object. The modification must occur due to the calling of an API or a subsequent node failure. For simple clusters, the rejoining node is updated with a copy of the CRG from some node that is currently active in the cluster. However, this may not be true in all cases.

See Starting or ending a cluster administrative domain node for information about the behavior of a cluster administrative domain during a rejoin.

***Example: Rejoin***

This topic describes the actions that can occur when a node rejoins an IBM i cluster.

The following diagram describes the actions taken whenever a node rejoins the cluster. In addition, the state of the rejoining nodes will be changed from *inactive* to *active* in the membership status field in the recovery domain of the CRG. The exit program is called on all nodes in the CRG's recovery domain and is passed an action code of Rejoin.

*Table 6. Rejoin operation*

| **Rejoin operation** | | | |
|---|---|---|---|
| Rejoining node | | Cluster nodes | |
| Contains copy of CRG | Does not contain copy of CRG | Contain copy of CRG | Do not contain copy of CRG |
| (1) | (2) | (3) | (4) |

Using the diagram above, the following situations are possible:

1. 1 and 3
2. 1 and 4
3. 2 and 3
4. 2 and 4

If a node in the cluster has a copy of the CRG, the general rule for rejoin is that the CRG is copied from an active node in the cluster to the rejoining node.

**Rejoin Situation 1**

A copy of the CRG object from a node in the cluster is sent to the joining node. The result is:

- The CRG object is updated on the joining node with the data sent from the cluster.
- The CRG object may be deleted from the joining node. This can occur if the joining node was removed from the CRG's recovery domain while the joining node was out of the cluster.

**Rejoin Situation 2**

A copy of the CRG object from the joining node is sent to all cluster nodes. The result is:

- No change if none of the cluster nodes are in the CRG's recovery domain.
- The CRG object may be created on one or more of the cluster nodes. This can occur in the following scenario:
  - Nodes A, B, C, and D make up a cluster.
  - All four nodes are in the recovery domain of the CRG.
  - While node A is out of the cluster, the CRG is modified to remove B from the recovery domain.
  - Nodes C and D fail.
  - The cluster is only node B which does not have a copy of the CRG.
  - Node A rejoins the cluster.
  - Node A has the CRG (although it is down level by now) and Node B does not. The CRG is created on node B. When nodes C and D rejoin the cluster, the copy of the CRG in the cluster updates node C and D and the previous change to remove node B from the recovery domain is lost.

**Rejoin Situation 3**

A copy of the CRG object from a node in the cluster is sent to the joining node. The result is:

- No change if the joining node is not in the CRG's recovery domain.
- The CRG object may be created on the joining node. This can occur if the CRG was deleted on the joining node while cluster resource services is not active on the node.

**Rejoin Situation 4**

Some internal information from one of the nodes in the cluster may be used to update information about the joining node but nothing occurs that is visible to you.

## Advanced node failure detection

Advanced node failure detection function is provided which can be used to reduce the number of failure scenarios which result in cluster partitions.

There are some failure situations in which heartbeat monitoring cannot determine what exactly failed. Failure might be the result of a communication failure between cluster nodes or an entire cluster node has failed. Take for example, the case where a cluster node fails due to a failure in a critical hardware component such as a processor. The whole machine can go down without giving cluster resource services on that node an opportunity to notify other cluster nodes of the failure. The other cluster nodes see only a failure in the heartbeat monitoring. They are unable to know if it was due to a node failure or a failure in some part of the communication path (a line, a router, or an adapter).

When this type of failure occurs, cluster resource service assumes that the node that is not responding could still be operational and partitions the cluster.

In 7.1, an advanced node failure detection function is provided which can be used to reduce the number of failure scenarios which result in cluster partitions. An additional monitoring technique is used to provide another source of information to allow cluster resource services to determine when a cluster node has failed.

This advanced function uses a Hardware Management Console (HMC) for those IBM systems which can be managed by an HMC or a Virtual I/O Server (VIOS) partition on an Integrated Virtualization Manager (IVM) managed server. In either case, the HMC or IVM is able to monitor the state of logical partitions or the entire system and notify cluster resource services when state changes in the partition or system occur. Cluster resource services can use this state change information to know when a cluster node has failed and avoid partitioning the cluster with only the knowledge of a heartbeat monitor.

In this example, an HMC is being used to manage two different IBM systems. For example, the HMC can power up each system or configure logical partitions on each system. In addition, the HMC is monitoring the state of each system and logical partitions on each system. Assume that each system is a cluster node and cluster resource services is monitoring a heartbeat between the two cluster nodes.

With the advanced node failure detection function, cluster resource services can be configured to make use of the HMC. For example, Node A can be configured to have a cluster monitor that uses the HMC. Whenever HMC detects that Node B fails (either the system or the logical partition for Node B), it notifies cluster resource services on Node A of the failure. Cluster resource services on Node A then marks Node B as failed and perform failover processing rather than partitioning the cluster.

Likewise, Node B can also be configured to have a cluster monitor. In this example, then, a failure of either Node A or Node B would result in a notification from the HMC to the other node.

Refer to Managing failover outage events for more information about failure scenarios that result in a cluster partition when advanced node failure detection is not used and that result in node failure when the advanced detection is used.

Notification of failures by an HMC with Common Information Model (CIM) server or VIOS depends upon a CIM server running on the cluster node which is to receive the notification. If the CIM server is not running, the advanced node failure detection is not aware of node failures. The CIM server must be started and left running anytime the cluster node is active. Use the STRTCPSVR *CIMOM CL command to start the CIM server.

If you are using an HMC with version V8R8.5.0 or later you can configure a cluster monitor that uses an HMC REST server. This type of cluster monitor does not require a CIM server.

## Cluster administrative domain

A *cluster administrative domain* provides a mechanism of maintaining a consistent operational environment across cluster nodes within an IBM i high availability environment. A cluster administrative domain ensures that highly available applications and data behave as expected when switched to or failed over to backup nodes.

There are often configuration parameters or data associated with applications and application data which are known collectively as the operational environment for the application. Examples of this type of data include user profiles used for accessing the application or its data, or system environment variables that control the behavior of the application. With a high availability environment, the operational environment needs to be the same on every system where the application can run, or where the application data resides. When a change is made to one or more configuration parameters or data on one system, the same change needs to be made on all systems. A cluster administrative domain lets you identify resources that need to be maintained consistently across the systems in an IBM i high availability environment. The cluster administrative domain then monitors for changes to these resources and synchronizes any changes across the active domain.

When a cluster administrative domain is created, the system creates a peer CRG with the same name. The nodes that make up the cluster administrative domain are defined by the CRGs recovery domain. Node membership of the cluster administrative domain can be modified by adding and removing nodes from the recovery domain using **`Add Cluster Admin Domain Node Entry (ADDCADNODE)`** and **`Remove Cluster Admin Domain Node Entry (RMVCADNODE)`** or by using the **`Work Cluster (WRKCLU) command`**. Each cluster node can be defined in only one cluster administrative domain within the cluster.

Once the cluster administrative domain is created, it can be managed with CL commands or the IBM PowerHA SystemMirror for i graphical interface in IBM Navigator for i.

**Note:** To work with PowerHA commands or the PowerHA graphical interface, you must have IBM PowerHA SystemMirror for i licensed program installed.

### Monitored resources

A *monitored resource* is a system resource that is managed by a cluster administrative domain. Changes made to a monitored resource are synchronized across nodes in the cluster administrative domain and applied to the resource on each active node. Monitored resources can be system objects such as user profiles or job descriptions. A monitored resource can also be a system resource not represented by a system object, such as a single system value or a system environment variable. These monitored resources are represented in the cluster administrative domain as *monitored resource entries (MREs)*.

Cluster administrative domain supports monitored resources with simple attributes and compound attributes. A compound attribute differs from a simple attribute in that it contains zero or more values, while a simple attribute contains a single value. Subsystem Descriptions (*SBSD) and Network Server Descriptions (*NWSD) are examples of monitored resources that contain compound attributes.

In order for MREs to be added, the resource must exist on the node from which the MREs are added. If the resource does not exist on every node in the administrative domain, the monitored resource is created. If a node is later added to the cluster administrative domain, the monitored resource is created. MREs cannot be added in the cluster administrative domain if the domain has status of Partitioned.

Determine the status of the cluster administrative domain and the status of the nodes in the domain by using PowerHA graphical interfaces in IBM Navigator for i, the **`Display CRG Information (DSPCRGINF)`** or **`Work with Cluster (WRKCLU)`** commands.

**Note:** To use the PowerHA graphical interface or the PowerHA commands, you must have the IBM PowerHA SystemMirror for i licensed program installed.

You can also use Cluster APIs to determine the status of a cluster administrative domain.

When an MRE is added to the cluster administrative domain, changes made to the resource on any active node in the cluster administrative domain are propagated to all nodes in the active domain. When a node within a cluster administrative domain is inactive, the synchronization option controls the way changes are propagated throughout the cluster. When the synchronization option is set to Active Domain, any changes made to the resource on the inactive node are discarded when the node rejoins the cluster. When the synchronization option is set to Last Change, changes made to the resource on the inactive node are only discarded if there was a more recent change to the resource propagated in the cluster administrative domain. When the cluster administrative domain is deleted, all monitored resource entries that are defined in the cluster administrative domain are removed; however, the actual resource is not removed from any node in the active domain.

To help you manage the administrative domain with many MREs, the **`Print Admin Domain MRE (PRTCADMRE)`** or **`Work with Monitored Resources (WRKCADMRE)`** command can be used. Information that can be printed or directed to a database output file and can be used to write additional tools, perform queries, or manage the monitored resource in the administrative domain. If your monitored resources ever have a global status of inconsistent, the PRTCADMRE or WRKCADMRE command helps you figure out which MRE(s) are not consistent and on what systems.

**Related information**

Attributes that can be monitored

Planning a cluster administrative domain

Creating a cluster administrative domain

Add monitored resource entries

Managing cluster administrative domain

Cluster APIs

PowerHA commands

## PowerHA data replication technologies

PowerHA offers several different technologies for data replication. These technologies can be used separately or sometimes combined to provide a higher level of protection against outages.

Geographic mirroring is an IBM i replication technology and can be used with any storage. Data is replicated between two copies of the independent ASP, and both synchronous and asynchronous replication is supported. Geographic mirroring can protect against both server and storage outages.

For customers with external storage, there are several technologies available. Switched logical units allow the switching of one copy of the data from one system to another and protect against server outages. Metro Mirror and Global Mirror are the synchronous and asynchronous replication technologies for external storage and protect against both server and storage outages by replicating the data from the primary copy of the IASP to a backup copy. HyperSwap is a DS8000 technology, which provides a near zero outage in the case of storage outages. FlashCopy® is a point in time copy mechanism, which can be used in combination with any of the technologies for backups and for other uses.

## Geographic mirroring

Geographic mirroring uses IBM i cluster technology to provide a high availability solution where a consistent copy of data that is stored in an independent disk pool at the production system is maintained on a mirrored copy. Geographic mirroring maintains a consistent backup copy of an independent disk pool by using internal or external storage.

If an outage occurs on the production site, production is switched to the backup site, which contains the mirrored copy of data, typically located at another location. In synchronous delivery mode, data is mirrored before write operations complete on the production system and is typically used for applications that cannot suffer any data loss in the event of a failure. Data is still sent to the mirror copy before the write operation completes while in asynchronous delivery mode, however, control is returned to the application before the mirrored write actually makes it to the mirror copy.

A good reason for using the existing synchronous delivery mode, is if the application wants to make sure that all completed writes on the production side have made it to the mirror copy side.

Geographic mirroring provides logical page level mirroring between independent disk pools through the use of data port services. Data port services manages connections for multiple IP addresses, which provides redundancy and greater bandwidth in geographic mirroring environments.

Geographic mirroring allows for the production and mirrored copies to be separated geographically, which allows for protection in the event of a site-wide outage. When planning a geographic mirroring solution, the distance between production and mirrored independent disk pools might affect the application response time. Longer distances between the production and mirrored copies might cause longer response times. Before implementing a high availability solution that uses geographic mirroring, you must understand your distance requirements and the associated performance implications on your applications.

With asynchronous delivery mode, the application response times are not impacted as with synchronous delivery mode. Large amounts of latency can result in additional main storage and CPU resources needed for asynchronous delivery mode.

Geographic mirroring is a technology available with IBM PowerHA SystemMirror for i.

Geographic mirroring with asynchronous delivery mode is only available with PowerHA version 2.0 and above.

**Related information**

Planning geographic mirroring

Configuring geographic mirroring

Managing geographic mirroring

Scenario: Geographic mirroring

## Metro Mirror

You can configure an IBM i high availability solution that uses Metro Mirror. Metro Mirror maintains a consistent copy of data between two IBM System Storage® external storage units.

Metro Mirror, when used with cluster technology, provides a high availability and disaster recovery solution. Like geographic mirroring, this technology also mirrors data stored in independent disk pools. However, with Metro Mirror, disks are located on IBM System Storage external storage units. Mirroring occurs from the source external storage units, which are usually located at the production site, to a set of target storage units, which are usually at the backup site. The data is copied between the external storage units to provide availability for both planned and unplanned outages.

Metro Mirror is a function of the external storage unit where the target copy of a volume is constantly being updated to match changes that are made to a source volume. It is typically used for applications that cannot suffer any data loss in the event of a failure. The source and target volumes can be on the same external storage unit or on separate external storage unit. In the case of separate units, the target storage unit can be located at another site up to 300 kilometers (186 miles) away. However, there might be performance implications when using synchronous communications over this distance and it might be more practical to consider one shorter to minimize the performance impact.

**Related information**

Planning Metro Mirror

Configuring Metro Mirror

Managing Metro Mirror

Scenario: Metro Mirror

PowerHA supported storage servers

## Global Mirror

You can configure a IBM i high availability solution that uses Global Mirror. Global Mirror maintains a consistent copy of data between two IBM System Storage external storage units.

Global Mirror provides disk I/O subsystem level mirroring between two external storage units. This asynchronous solution provides better performance at unlimited distances, by allowing the target site to trail in currency a few seconds behind the source site.

Global Mirror provides a remote, long-distance remote copy across two sites using asynchronous technology. It operates over high-speed, fibre channel communication links and is designed to maintain a complete and consistent remote mirror of data asynchronously at virtually unlimited distances with almost no impact to application response time.

Separating data centers by longer distances helps to provide protection from regional outages. This asynchronous technique provides better performance at unlimited distances. With Global Mirror, data that is copied to the backup site is current with the production site in a matter of seconds. Global Mirror creates a disaster recovery solution that provides high performance and a cost effective approach to data replication across global distances.

**Related information**

Planning Global Mirror

Configuring Global Mirror

Managing Global Mirror

Scenario: Global Mirror

## Switched logical units

A *switched logical unit* is an independent disk pool. When switched logical units are combined with IBM i clusters technology, you can create a simple and cost effective high availability solution for planned and some unplanned outages.

The device cluster resource group (CRG) controls the independent disk pool, which can be switched automatically in the case of an unplanned outage, or it can be switched manually by a switchover.

A group of systems in a cluster can take advantage of the switchover capability to move access to the switched logical unit pool from system to system. A switchable logical unit must be in an IBM System Storage connected through a storage area network. When the independent disk pool is switched the logical units within the IBM System Storage unit are reassigned from one system to another system. See PowerHA supported storage server, for the storage technologies that support switchable logical units.

**Related information**

Planning switched logical units (LUNs)

Configuring switched logical units (LUNs)

Managing switched logical units (LUNs)

PowerHA supported storage servers

## FlashCopy

In IBM i high availability environments that use IBM System Storage external storage units, you can use FlashCopy. FlashCopy provides an almost instant, point-in-time copy of independent disk pools on external storage, which can reduce the time it takes to complete daily backups.

Point-in-time copy functions give you an instantaneous copy, or view, of what the original data looked like at a specific point-in-time. The target copy is totally independent of the source independent disk pool and is available for both read and write access once the FlashCopy command has been processed.

**Related information**

Planning FlashCopy

Configuring a FlashCopy session

Managing the FlashCopy technology

Scenario: Performing a FlashCopy function

## DS8000 Full System HyperSwap

In IBM i high availability environments, use full system HyperSwap as a means to help reduce or eliminate outages due to storage and SAN-related outages.

Full system HyperSwap is a single system IBM i storage hardware high availability solution that uses Metro Mirror on IBM Systems Storage DS8000 devices to maintain a consistent copy of the data between two IBM Systems Storage external storage units. Full system HyperSwap allows for a planned or unplanned IBM Systems Storage external storage device switchover to occur without taking applications offline for the switchover. Full system HyperSwap supports full system (SYSBAS) replication only, and does not support independent disk pool replication.

Full system HyperSwap has the same distance limitations as a traditional Metro Mirror cross site mirroring solution. The source and target volumes can be on the same external storage unit or on separate external storage units. In the case of separate units, the target storage unit can be located at another site up to 300 kilometers (186 miles) away. However, there might be performance implications when using synchronous communications over this distance and it might be more practical to consider one shorter to minimize the performance impact.

To use full system HyperSwap, you must have IBM PowerHA for i Express® Edition installed on your system. To use full system HyperSwap does not require a cluster and does not use cluster technologies.

**Related information**

Planning for DS8000 Full System HyperSwap

Configuring DS8000 Full System HyperSwap

Managing DS8000 Full System HyperSwap

## DS8000 HyperSwap with independent auxiliary storage pools (IASPs)

In IBM i high availability environments, use HyperSwap as a means to help reduce or eliminate outages due to storage and SAN-related outages.

HyperSwap is a storage high availability solution that allows for logical units that are mirrored between two IBM System Storage DS8000 units to be switched with a near zero outage time. When HyperSwap is implemented at an IASP level, it can be combined with other PowerHA technologies to provide a minimal downtime solution for planned and unplanned storage outages, and a minimal downtime solution for server planned and unplanned outages.

To use HyperSwap, you must have IBM PowerHA for i Enterprise Edition installed on your system. To use DS8000 HyperSwap with IASPs does require a cluster and does use PowerHA technologies.

**Related information**

Planning DS8000 HyperSwap with independent auxiliary storage pools (IASPs)

Configuring DS8000 HyperSwap with independent auxiliary storage pools (IASPs)

Managing DS8000 HyperSwap with independent auxiliary storage pools (IASPs)

## High availability management

To plan, configure, and manage a complete high availability solution requires a set of management tools and offerings. With IBM i systems, several choices exist for high availability management.

Depending on your needs and requirements, high availability management provides graphical interfaces, commands, and APIs that can be used to create and manage your environment. You can also choose to use an IBM business partner application. Each of these choices of high availability management tools has their advantages and limitations.

### IBM PowerHA SystemMirror for i interfaces

IBM PowerHA SystemMirror for i, licensed program number (5770-HAS), is an end-to-end high availability offering. When combined with independent auxiliary storage pools (iASPs) and HA Switchable Resources (HASR - Option 41), it enables a complete solution to be deployed via IBM System Storage server or internal disk. PowerHA provides several interfaces to configure and manage high availability solutions and technology.

For more information on the storage technologies that are provided by IBM i, see PowerHA supported storage servers.

The IBM PowerHA SystemMirror for i licensed program provides a graphical interface that allows you to configure and manage a high availability solution. This product also provides corresponding commands and APIs for functions that are related to high availability technologies. With this licensed program, high availability administrators can create and manage a high availability solution to meet their business needs, using interfaces that fit their skills and preferences. You can also work with multiple interfaces interchangeably, using graphical interfaces for some tasks and commands and APIs for others.

The IBM PowerHA SystemMirror for i licensed program provides the following interfaces:

**PowerHA graphical interface**

This graphical interface allows you to easily configure, monitor, and manage your High Availability solution. For customers upgrading from a release prior to 7.2, it combines the simplicity of the High Availability Solutions Manager graphical interface with the flexibility of the Cluster Resource Services graphical interface in a single graphical interface.

**PowerHA commands**

These commands allow you to configure and manage your high availability solution via a command-line interface.

**PowerHA APIs**

These APIs allow you to work with PowerHA version, and retrieve PowerHA related information.

**Related information**

Installing IBM PowerHA SystemMirror for i licensed program

Cluster APIs

PowerHA APIs

IBM PowerHA for i commands

PowerHA supported storage servers

#### PowerHA graphical interface

The IBM PowerHA SystemMirror for i licensed program provides a graphical interface that allows you to do tasks with IBM i high availability technologies to configure, monitor, and manage a high availability solution.

You can use the PowerHA graphical interface to create and manage clusters, cluster resource groups, device domains, cluster administrative domains, and independent ASPs, all from a single graphical interface.

### PowerHA commands

IBM PowerHA SystemMirror for i licensed program provides IBM i command line interfaces to configure and manage your high availability solution.

The PowerHA commands consist of the following categories:

- Cluster administrative domain commands
- Monitored resource entries commands
- Cluster commands
- Independent disk pools commands
  - Copy description commands
  - Session commands
  - HyperSwap commands
  - HA configuration description commands

**Related information**

IBM PowerHA for i commands

### PowerHA APIs

IBM PowerHA SystemMirror for i licensed program provides APIs that can be used to implement IBM System Storage mirroring technologies and cross-site mirroring functions that can be used by IBM i application providers or customers to enhance their application availability.

To use these APIs, you must have the IBM PowerHA SystemMirror for i licensed program installed on your systems in your high availability environment. The following APIs are provided:

- `Change High Availability Version (QhaChangeHAVersion)` API
- `List High Availability Information (QhaListHAInfo)` API
- `Retrieve High Availability Information (QhaRetrieveHAInfo)` API
- `Retrieve ASP Copy Information (QyasRtvInf)` API
- `Retrieve HA Configuration Description (QhaRetrieveConfigurationDesc) API`
- `Retrieve HA Status (QhaRetrieveStatus) API`

**Related information**

Cluster APIs

## IBM PowerHA SystemMirror for i version support

A PowerHA version represents the level of PowerHA for i function available on a cluster.

### Enhanced IBM PowerHA SystemMirror for i licensed program number (5770-HAS)

In 7.2, IBM PowerHA SystemMirror for i licensed program was enhanced. More function was added to the graphical interface, command-line interface, and APIs. This new function can assist administrators in configuring and managing high availability solutions. Refer to the following topics for details on the features of each of these interfaces:

- PowerHA graphical interface
- PowerHA commands
- PowerHA APIs

### IBM PowerHA SystemMirror for i version support

A PowerHA version represents the level of IBM PowerHA SystemMirror for i function available on a cluster. It is displayed in the form, x.y. For example, version 2.0 is a valid PowerHA version.

There are two PowerHA versions:

**Potential PowerHA version**

Represents the version of IBM PowerHA SystemMirror for i installed on a node. The potential PowerHA is the highest PowerHA version the node can currently support. The potential PowerHA version is updated when a new version of IBM PowerHA SystemMirror for i is installed.

**Current PowerHA version**

Represents the version currently being used for all cluster operations. This is the version of communications between the nodes in the cluster. To take advantage of all available new PowerHA function, every node in the cluster needs to be at the latest potential PowerHA version and the current PowerHA version must be adjusted to match.

### Compatibility of PowerHA versions

IBM PowerHA SystemMirror for i supports up to a three version difference among nodes. Nodes with a potential PowerHA version of at least equal to the current PowerHA version, but not more than three versions higher than the current PowerHA version, are compatible. For example, if the current PowerHA version is 2.1, nodes with a potential PowerHA version of at least 2.1 and less than 6.0 are compatible.

### Setting the current PowerHA version

The current PowerHA version is initially set when the cluster is created with the Create Cluster (CRTCLU) command. If the cluster exists when IBM PowerHA SystemMirror for i is installed, the current PowerHA version is set to lowest current PowerHA version supported by the node. Adjusting the current PowerHA version is done with the Change Cluster Version (CHGCLUVER) command. The current PowerHA version can only be adjusted if the potential PowerHA version of each node in the cluster supports the version. The current PowerHA version cannot be changed back to a lower version. Once the current PowerHA version is adjusted to a higher version, new IBM PowerHA SystemMirror for i function is made available.

The cluster version identifies the communication level of the nodes in the cluster, the ability of a node to join the cluster and the ability of the cluster to support new function. Refer to the information about compatibility between cluster versions and IBM I series releases to verify your PowerHA, clusters and operating system are compatible. Cluster Version

### Summary of new functions by PowerHA version

It is implied that functions available include functions available at previous versions.

Function available with PowerHA version 1.0, requires cluster version 6.0.

New function available with PowerHA version 2.0, requires cluster version 7.0.

#### Functions available with PowerHA version 1.0

Functions available with PowerHA version 1.0, which requires current cluster version 6, are:

Support for Cross Site Mirroring (XSM):

- Geographic Mirroring
- Metro Mirror
- Global Mirror
- FlashCopy

IBM PowerHA SystemMirror for i commands:

- Add ASP Copy Description (ADDASPCPYD)
- Add Admin Domain MRE (ADDCADMRE)
- Add Admin Domain Node Entry (ADDCADNODE)
- Add Cluster Node Entry (ADDCLUNODE)
- Add CRG Device Entry (ADDCRGDEVE)
- Add CRG Node Entry (ADDCRGNODE)
- Add Device Domain Entry (ADDDEVDMNE)
- Change ASP Copy Description (CHGASPCPYD)

- Change ASP Session (CHGASPSSN)
- Change Cluster Admin Domain (CHGCAD)
- Change Cluster (CHGCLU)
- Change Cluster Node Entry (CHGCLUNODE)
- Change Cluster Version (CHGCLUVER)
- Change Cluster Resource Group (CHGCRG)
- Change CRG Device Entry (CHGCRGDEVE)
- Change CRG Primary (CHGCRGPRI)
- Create Cluster Admin Domain (CRTCAD)
- Create Cluster (CRTCLU)
- Create Cluster Resource Group (CRTCRG)
- Delete Cluster Admin Domain (DLTCAD)
- Delete Cluster (DLTCLU)
- Delete CRG Cluster (DLTCRGCLU)
- Display ASP Copy Description (DSPASPCPYD)
- Display ASP Session (DSPASPSSN)
- Display Cluster Information (DSPCLUINF)
- Display CRG Information (DSPCRGINF)
- End ASP Session (ENDASPSSN)
- End Cluster Admin Domain (ENDCAD)
- End Cluster Node (ENDCLUNOD)
- End Cluster Resource Group (ENDCRG)
- Remove ASP Copy Description (RMVASPCPYD)
- Remove Admin Domain MRE (RMVCADMRE)
- Remove Admin Domain Node Entry (RMVCADNODE)
- Remove Cluster Node Entry (RMVCLUNODE)
- Remove CRG Device Entry (RMVCRGDEVE)
- Remove CRG Node Entry (RMVCRGNODE)
- Remove Device Domain Entry (RMVDEVDMNE)
- Start ASP Session (STRASPSSN)
- Start Cluster Admin Domain (STRCAD)
- Start Cluster Node (STRCLUNOD)
- Start Cluster Resource Group (STRCRG)
- Work with ASP Copy Description (WRKASPCPYD)
- Work with Cluster (WRKCLU)

IBM PowerHA SystemMirror for i application programming interfaces:

- Change Device Domain Data (QYASCHGDDD)
- Retrieve ASP Copy Information (QYASRTVINF)
- Retrieve Device Domain Data (QYASRTVDDD)

IBM PowerHA SystemMirror for i graphical user interfaces:

- Cluster Resource Services
- High Availability Solutions Manager

### Functions available with PowerHA version 2.0

Functions available with PowerHA version 2.0, which requires current cluster version 7 are:

**Support for PowerHA versions:**

- Enhanced Change Cluster Version (CHGCLUVER) command
- New Change High Availability Version (QhaChangeHAVersion) API
- Enhanced Create Cluster (CRTCLU) command
- Enhanced Display Cluster Information (DSPCLUINF) command
- New List High Availability Information (QhaListHAInfo) API
- NewRetrieve Cluster (RTVCLU) command
- NewRetrieve Cluster Resource Group (RTVCRG) command
- New Retrieve High Availability Information (QhaRetrieveHAInfo) API
- Enhanced Work with Cluster (WRKCLU) command

**Support for asynchronous geographic mirroring:**

- Enhanced Change ASP Session (CHGASPSSN) command
- Enhanced Display ASP Session (DSPASPSSN) command
- Enhanced Retrieve ASP Session (RTVASPSSN) command
- Enhanced Retrieve ASP Copy Description (RTVASPCPYD) command

**Support for enhanced cluster node failure detection:**

- New Add Cluster Monitor (ADDCLUMON) command
- New Change Cluster Monitor (CHGCLUMON) command
- New Remove Cluster Monitor (RMVCLUMON) command

**IPv6 support added**

The following functions have been enhanced and now support IPv6.

- Enhanced Add Cluster Node Entry (ADDCLUNODE) command
- Enhanced Add Cluster Resource Group Device Entry (ADDCRGDEVE) command
- Enhanced Add Cluster Resource Group Node Entry (ADDCRGNODE) command
- Enhanced Change Cluster Node Entry (CHGCLUNODE) command
- Enhanced Change Cluster Resource Group (CHGCRG) command
- Enhanced Change Cluster Resource Group Device Entry (CHGCRGDEVE) command
- Enhanced Create Cluster (CRTCLU) command
- Enhanced Create Cluster Resource Group (CRTCRG) command
- Enhanced Display Cluster Information (DSPCLUINF) command
- Enhanced Display Cluster Resource Group Information (DSPCRGINF) command
- Enhanced Work with Cluster (WRKCLU) command

**Support for printer devices and authorization lists in Cluster Administrative Domain:**

- Enhanced Add Admin Domain MRE (ADDCADMRE) command
- New Print Cluster Admin Domain MRE (PRTCADMRE) command
- Enhanced Remove Admin Domain MRE (RMVCADMRE) command

**Support for switched logical units**

- Support is now available for *switched logical units.* See the Switched logical units topic for additional information.

### PowerHA server job

When clustering is active and the current PowerHA version is 2.0 or higher, the PowerHA server job will be running. This job is named QHASVR and runs in the QSYSWRK subsystem under the QHAUSRPRF user profile.

### Functions available with PowerHA version 2.1

Functions available with PowerHA version 2.1, which requires current cluster version 7 are:

- PowerHA graphical interface
- New Work with Monitored Resources (WRKCADMRE) command

### Functions available with PowerHA version 2.2

Functions available with PowerHA version 2.2, which requires current cluster version 7 are:

- All DS8000, SAN Volume Controller, and Storwize® V7000 and V3700 PowerHA functionality is now available through the new PowerHA GUI, which is available under IBM Navigator for i.
- Enhanced command interface, Work with Monitored Resources Entries (MRE) (WRKCADMRE) has been created, offering improved sorting, filtering, and searching capabilities for large number of MREs.
- Administrative domain entries can be added or removed when not all nodes are active in the cluster.
- Administrative domain, FlashCopy, and copy description functions are now fully cluster enabled so that all work can be initiated from any active cluster node.
- Geographic mirroring synchronization priority can be changed while the IASP is varied on.
- PowerHA submitted jobs are now associated with the QCSTJOBD job description instead of the QBATCH job description.

### Functions available with PowerHA version 3.0

Functions available with PowerHA version 3.0, which requires current cluster version 8 are:

- Cluster administrative domain monitored resource limit raised to 45,000. So cluster administrative domain can synchronize 45,000 specific objects.
- Cluster administrative domain with IBM PowerHA SystemMirror for i now supports synchronization of object authority and ownership. New object attributes of object owner, authority entry, authorization list and primary group were added to some monitored resource entries.
- IBM PowerHA for i Express Edition edition will support DS8000 HyperSwap. The following commands have been added to support HyperSwap:
  - New Add HyperSwap Storage Description command
  - New Change HyperSwap Storage Description command
  - New Change HyperSwap Status command
  - New Display HyperSwap Storage Description command
  - New Display HyperSwap Status command
  - New Remove HyperSwap Storage Description command
- The High Availability Solutions Manager graphical interface and Cluster Resource Services graphical interface are no longer supported. Use the PowerHA graphical interface.

### Functions available with PowerHA version 3.1

Functions available with PowerHA version 3.1, which requires current cluster version 8 are:

- IBM PowerHA for i Enterprise Edition edition supports DS8000 HyperSwap with IASPs. The following commands have been added to support HyperSwap with IASPs:
  - Updated Add ASP Copy Description command

- Updated Change ASP Copy Description command
- Updated Retrieve ASP Copy Description command
- Updated Retrieve ASP Session command
- Updated Change HyperSwap Status command
- Updated Display HyperSwap Status command
- New Work HyperSwap StatusWork with HyperSwap Status command
- New Add HA Configuration DescriptionAdd HA Configuration Description command
- New Remove HA Configuration DescriptionRemove HA Configuration Description command
- New Display HA Configuration DescriptionDisplay HA Configuration Description command
- New Work HA Configuration DescriptionWork HA Configuration Description command
- New Retrieve SAN Volume Controller (SVC) Copy DescriptionRetrieve SAN Volume Controller Copy Description command
- New Retrieve SVC SessionRetrieve SAN Volume Controller Session command

- IBM PowerHA for i Enterprise Edition edition supports DS8000 HyperSwap with IASPs. The following APIs have been added to support HyperSwap with IASPs:
  - Updated Retrieve ASP Copy Information API
  - New Retrieve HA Configuration DescriptionRetrieve HA Configuration Description API
  - New Retrieve HA StatusRetrieve HA Status API

**Functions available with PowerHA version 4.0**

## Option 41 (HA Switchable Resources)

Option 41 (HA Switchable Resources) is required when using several IBM i high availability management interfaces and functions require its installation in order to be used.

Option 41 (High Availability Switchable Resources) is required if you plan to use the following interfaces:

- IBM PowerHA SystemMirror for i licensed program.
  - PowerHA graphical interface
  - PowerHA commands
  - PowerHA APIs

Option 41 is also required to create or work with a device domain.

## High availability function in the base operating system

Some cluster CL commands and all Cluster APIs exist in the base IBM i.

### Cluster commands

The following cluster commands will remain in QSYS for debugging purposes and for deleting cluster-related objects:

- Delete Cluster Resource Group (DLTCRG) command
- Dump Cluster Trace (DMPCLUTRC) command
- Change Cluster Recovery (CHGCLURCY) command
- Start Clustered Hash Table Server (STRCHTSVR) command
- End Clustered Hash Table Server (ENDCHTSVR) command

### Cluster APIs

You can write your own custom application to configure and manage your cluster by using Cluster APIs. These APIs take advantage of the technology provided by cluster resource services provided as a part of

IBM i. New enhanced functions are included in the IBM PowerHA SystemMirror for i commands which are provided by the IBM PowerHA SystemMirror for i licensed program.

**Related information**

Cluster APIs

## Related information for High availability technologies

Product manuals, IBM Redbooks® publications, websites, and other information center topic collections contain information that relates to the High availability topic collection. You also can find related information about implementing independent disk pools, PowerHA technologies, and disaster recovery. You can view or print any of the PDF files.

### IBM Redbooks

- IBM i and IBM Storwize Family: A Practical Guide to Usage Scenarios
- IBM i and IBM System Storage: A Guide to Implementing External Disks on IBM i
- Implementing PowerHA for IBM i
- Introduction to Storage Area Networks
- iSeries in Storage Area Networks: A Guide to Implementing FC Disk and Tape with iSeries
- PowerHA SystemMirror for IBM i Cookbook
- Simple Configuration Example for Storwize V7000 FlashCopy and PowerHA SystemMirror for i

### Websites

- High availability with IBM PowerHA This is the IBM site for High Availability and Clusters for i, UNIX, and Linux.
- IBM i
- IBM PowerHA SystemMirror for i
- IBM PowerHA SystemMirror for i wiki
- IBM Storage
- Power® Services

  This IBM site contains the Systems lab services and training that is offered for IBM i.
- IBM System Storage Interoperation Center (SSIC)
- IBM Techdocs Library

  This site gives you access to the most current installation, planning, and technical support information available from IBM pre-sales support, and is constantly updated. Look for the latest and greatest technical documents on high availability, independent disk pools, SAP, JD Edwards, etc.
- Learning Services US

  This is the IBM site for IT product training, custom solutions, and e-Learning. You can search for courses that are offered on clustering and independent disk pools.
- Performance Management on IBM i
- Recommended fixes

This site provides links to available PTFs for several IBM i products. For PTFs related to high availability, select the topic High Availability: Cluster, IASP, XSM, and Journal.

#### Information Center topic collections

- Availability roadmap
- High availability overview
- High availability technologies
- IBM SAN Volume Controller Knowledge Center
- IBM Storwize V7000 Knowledge Center
- IBM Storwize V3700 Knowledge Center
- IBM DS8000 Knowledge Center
- Implementing high availability

#### Other information

- Disk management
- Resource Monitoring and Control (RMC)

## Resource Monitoring and Control (RMC)

Resource Monitoring and Control (RMC) is a generalized framework for managing, monitoring, and manipulating resources such as physical or logical system entities.

RMC is utilized as a communication mechanism for reporting service events to the Hardware Management Console (HMC). If RMC is not active, then service events will not be reported to the HMC. The following list describes services that are associated with RMC:

**CAS Daemon**

**Purpose:** Acts as the authentication server for RMC.

**Job Name:** QRMCCTCASD

**RMC Daemon**

**Purpose:** Monitors of resources by communicating with the Resources Managers.

**Job Name:** QRMCCTRMCD

**SRC Daemon**

**Purpose:**Monitors the status of the other RMC jobs; it will restart a job if that particular job unexpectedly ends.

**Job Name:** QRMCSRCD

#### Resource Managers (RM)

A Resource Manager (RM) is a job that manages and provides the interface between RMC and actual physical or logical entities. Although RMC provides the basic abstractions, such as resource classes, resources, and attributes for representing physical or logical entities, it does not itself represent any actual entities. An RM maps actual entities to RMC's abstractions. The following list describes the different Resource Managers that are supported for RMC:

**Audit Log RM**

**Purpose:** Provides a facility for recording information about the system's operation.

**Job Name:** QYUSALRMD

**CSMAgent RM**

**Purpose:** Provides resource classes to represent the Management Server, which is the HMC.

**Job Name:** QYUSCMCRMD

**Host RM**

**Purpose:** Provides resource classes to represent an individual machine.

**Job Name:** QRMCCTHRMD

**Service RM**

**Purpose:** Manages problem information and prepares it for delivery to the HMC.

**Job Name:** QSVRMSERMD

### Starting or ending the RMC

All RMC jobs, including RM jobs are in the QSYSWRK subsystem and are automatically started when the subsystem is started. TCP/IP must be active for startup to complete. The RMC Daemon requires TCP/IP to be active. If TCP/IP becomes inactive, then the RMC Daemon will end. The RMC Daemon will be automatically restarted by the SRC Daemon once TCP/IP becomes active again. No steps are required of the user under normal conditions. If RMC needs to be manually started, run the following command:

```
SBMJOB CMD(CALL PGM(QSYS/QRMCCTSRCD)) JOBD(QSYS/QRMCSRCD) PRTDEV(*JOBD) OUTQ(*JOBD)
USER(*JOBD) PRTTXT(*JOBD) RTGDTA(RUNPTY50)
```

If RMC needs to be manually ended, use the ENDJOB command to end the QRMCSRCD job. This command should end all RMC jobs. If all the jobs do not end, then manually end each of the jobs listed above.

# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

## Programming interface information

This Installing, upgrading, or deleting IBM i and related software publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i.

## Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java™ and all Java-based trademarks and logos are trademarks of Oracle, Inc. in the United States, other countries, or both.

Other product and service names might be trademarks of IBM or other companies.

## Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

**Personal Use:** You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

**Commercial Use:** You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM®