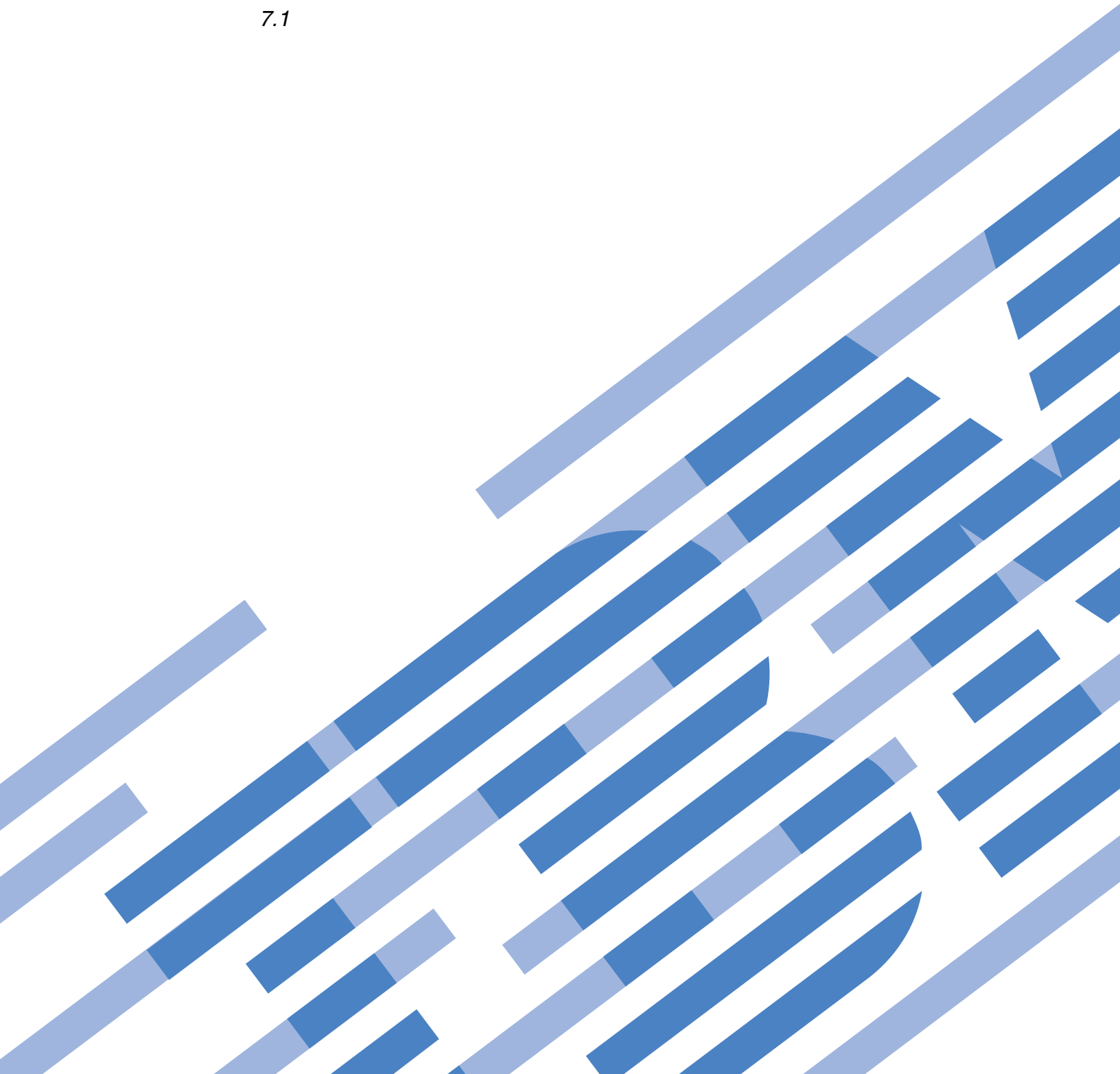
IBM i

Availability
Implementing High Availability with the task-based approach

7.1



IBM i

Availability
Implementing High Availability with the task-based approach

7.1

Note

Before using this information and the product it supports, read the information in "Notices," on page 197.

This edition applies to IBM i 7.1 (product number 5770-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

Contents

Chapter 1. Implementing high availability with a task-based approach

The task-based approach to configuring and managing i5/OS high availability allows you to configure and manage a customized high-availability solution based on your business needs. Graphical and command-line interfaces are used for configuring and managing your high-availability solution.

Unlike the solution-based approach, which uses the High Availability Solution Manager graphical interface, where a predefined solution is configured automatically with limited user input, the task-based approach gives the knowledgeable user the means to customize and implement a personalized solution. However, to create and manage a high-availability solution with this approach, users need to have a good understanding of their high availability needs and be familiar with several interfaces.

Cluster Resource Services graphical interface

The Cluster Resource Services interface allows you to configure and manage cluster technologies, which are integral to a high-availability solution. To use this interface the IBM® PowerHA® for i (iHASM) licensed program 5770-HAS installed. With this interface you can perform the following functions:

- Create and manage a cluster
- Create and manage nodes
- Create and manage cluster resource groups
- Create and manage cluster administrative domains
- Create and manage monitored resources
- Monitor the cluster for cluster related events, such as cluster partitions and failovers
- Perform manual switchovers for planned outages, such as scheduled maintenance to a system

Disk Management interface

The Disk Management interface allows you to configure and manage independent disk pools which are necessary when implementing several data resiliency technologies. Depending on the type of data resiliency technology that is implemented, installation requirements might be necessary to use some of these functions:

- Create a disk pool
- Make a disk pool available
- Make a disk pool unavailable
- Configure geographic mirroring
- Configure metro mirror
- Configure global mirror

Command line interface

The command line interface allows you to perform many different high availability tasks with CL commands. For each cluster-related task the corresponding CL command has been identified.

Related information:

IBM PowerHA for i commands

Planning your high availability solution

Before configuring an i5/OS high-availability solution, proper planning is necessary to ensure all the requirements for the solution have been met.

Each high availability technology has minimum requirements that should be met before configuring a specific solution. In addition to these requirements, it is important to also determine which resources should be made resilient. These resources, such as applications, data, and devices, should be evaluated to determine whether they should be highly available. If they require high availability, it is important to make any necessary changes to the environment prior to configuring a solution for high availability. For example, you might have data that resides in SYSBAS, which should be highly available. Before configuring a solution, you should move that data to an independent disk pool. Applications might also require changes to enable high availability.

Cluster applications

Application resilience is one of the key elements in a clustered environment. If you are planning to write and use highly available applications in your cluster you should be aware that these applications have specific availability specifications.

By taking advantage of resilient applications in your cluster, an application can be restarted on a different cluster node without requiring you to reconfigure the clients. In addition, the data that is associated with the application will be available after switchover or failover. This means that the application user can experience minimal, or even seamless, interruption while the application and its data switch from the primary node to the backup node. The user does not need to know that the application and data have moved on the back end.

In order to achieve application resiliency in your cluster, applications that meet certain availability specifications must be used. Certain characteristics must be present in the application in order for it to be switchable, and therefore always available to the users of the application in the cluster. See High Availability and Clusters for details on these application traits. Because these requirements exist, you have the following options for using a switchable application in your cluster:

1. **Purchase a cluster-enabled software application**

   Software products that are cluster-enabled meet certain high-availability requirements.
2. **Write or change your own application to make it highly available**

   Independent software vendors and application programmers can customize applications to allow them to be switchable in a System i® clustered environment.

Once you have a resilient application, it must be managed within your cluster.

Related information:

High Availability and Clusters

Identifying resilient applications

Not every application will give you the availability benefits of clustering.

An application must be resilient in order to take advantage of the switchover and failover capabilities provided by clustering. Application resilience allows the application to be restarted on the backup node without having to reconfigure the clients using the application. Therefore your application must meet certain requirements to take full advantage of the capabilities offered by clustering.

IBM i architecture for cluster-enabled applications

Additional end-user value is provided by any application that is highly available, recognizing applications that continue to be available in the event of an outage, planned or unplanned.

IBM i has provided an application resilience architecture that supports various degrees of highly available application. Applications on the high end of this spectrum demonstrate highly available characteristics, provide automation of the highly available environment, and are managed through high availability management interfaces.

These applications have the following characteristics:

- The application can switch over to a backup cluster node when the primary node becomes unavailable.
- The application defines the resilient environment in the Resilient Definition and Status Data Area to enable automatic configuration and activation of the application by a cluster management application.
- The application provides application resilience by means of an application CRG exit program to handle cluster related events, taking advantage of the capabilities of the IBM i cluster resource services.
- The application provides an application restart function that repositions the user to an application menu screen or beyond.

Applications that demonstrate more stringent availability and restart characteristics have the following characteristics:

- The application provides enhanced application resilience through more robust handling of cluster events (action codes) by the application CRG exit program.
- The application provides a greater level of application restart support. For host-centric applications, the user will be repositioned to a transaction boundary by commitment control or checkpoint functions. For client-centric applications, the user will experience a seamless failover with minimal service interruption.

Writing a highly available cluster application

A highly available application is one that can be resilient to a system outage in a clustered environment.

Several levels of application availability are possible:

1. If an application error occurs, the application restarts itself on the same node and corrects any potential cause for error (such as corrupt control data). You can view the application as though it had started for the first time.
2. The application performs some amount of checkpoint-restart processing. You can view the application as if it were close to the point of failure.
3. If a system outage occurs, the application is restarted on a backup server. You can view the application as though it had started for the first time.
4. If a system outage occurs, the application is restarted on a backup server and performs some amount of checkpoint-restart processing across the servers. You can view the application as if it were close to the point of failure.
5. If a system outage occurs, a coordinated failover of both the application and its associated data to another node or nodes in the cluster occurs. You can view the application as though it had started for the first time.
6. If a system outage occurs, a coordinated failover of both the application and its associated data to another node or nodes in the cluster occurs. The application performs some amount of checkpoint-restart processing across the servers. You can view the application as if it were close to the point of failure.

   Note: In cases 1 through 4 above, you are responsible for recovering the data.

Making application programs resilient:

Learn how to make application programs resilient.

A resilient application is expected to have the following characteristics:

- The application can be restarted on this node or another node

- The application is accessible to the client through the IP address
- The application is stateless or state information is known
- Data that is associated with the application is available after switchover

The three essential elements that make an application resilient to system outages in a clustered environment are:

The application itself
How tolerant is the application to errors or to system outages, and how transparently can the application restart itself?

The application can handle this through the use of clustering capabilities.

Associated data
When an outage occurs, does it affect the availability of any associated data?

You can store critical data in switched disks which allow data to remain available during an outage. Alternatively, a cluster middleware IBM Business Partner replication product that takes advantage of the clustering capabilities can handle this.

Control capabilities and administration
How easy is it to define the environment that supports the availability of the data and the application?

IBM PowerHA for i licensed program number, provides several interfaces to configure and manage high-availability solutions and technology. The PowerHA licensed program provides the following interfaces:

High Availability Solutions Manager graphical interface
This graphical interface allows you to select from several IBM i supported high-availability solutions. This interface validates all technology requirements for your selected solution, configures your selected solution and the associated technologies and provides simplified management of all the high availability technologies that comprise your solution.

Cluster Resource Services graphical interface
This graphical interface provides an experienced user more flexibility in customizing a high availability-solution. It allows you to configure and manage cluster technologies, such as CRGs. You can also configure some independent disk pools from this interface when they are used as part of a high availability solution.

IBM PowerHA for i commands
These commands provide similar functions but are available through a command-line interface.

APIs These IBM PowerHA for i APIs allow you to work with new function for independent disk pools.

In addition, you can also use a third-party cluster management interface that uses the clustering APIs and also combines resilient applications with resilient data can handle this.

Related information:

High availability management

Restarting highly available cluster applications:

To restart an application, the application needs to know its state at the time of the failover or switchover.

State information is application specific; therefore, the application must determine what information is needed. Without any state information, the application can be restarted on your PC. However, you must reestablish your position within the application.

Several methods are available to save application state information for the backup system. Each application needs to determine which method works best for it.

- The application can transfer all state information to the requesting client system. When a switchover or failover occurs, the application uses the stored state on the client to reestablish the state in the new server. This can be accomplished by using the Distribute Information API or Clustered Hash Table APIs.
- The application can replicate state information (such as job information and other control structures that are associated with the application) on a real-time basis. For every change in the structures, the application sends the change over to the backup system.
- The application can store pertinent state information that is associated with it in the exit program data portion of the cluster resource group for that application. This method assumes that a small amount of state information is required. You can use the Change Cluster Resource Group (QcstChangeClusterResourceGroup) API to do this.
- The application can store state information in a data object that is being replicated to the backup systems along with the application's data.
- The application can store state information in a data object contained in the switchable IASP that also contains the application's data.
- The application can store the state information about the client.
- No state information is saved, and you need to perform the recovery.

Note: The amount of information that is required to be saved is lessened if the application uses some form of checkpoint-restart processing. State information is only saved at predetermined application checkpoints. A restart takes you back to the last known checkpoint which is similar to how database's commitment control processing works.

Calling a cluster resource group exit program:

The cluster resource group exit program is called during different phases of a cluster environment.

This program establishes the environment necessary resiliency for resources within a cluster. 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 the following:

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

The exit program completes the following processes:

- Runs in a named activation group or the caller's activation group (*CALLER).
- Ignores the restart parameter if the exit program has an unhandled exception or is canceled.
- Provides a cancel handler.

When a cluster resource group API is run, the exit program is called from a separate job with the user profile specified on the Create Cluster Resource Group (QcstCreateClusterResourceGroup) API. The separate job is automatically created by the API when the exit program is called. If the exit program for a data CRG is unsuccessful or ends abnormally, the cluster resource group exit program is called on all active nodes in the recovery domain by using an action code of Undo. This action code allows any unfinished activity to be backed out and the original state of the cluster resource group to be recovered.

Suppose an unsuccessful switchover occurs for a device CRG. After switching back all the devices, if all of the devices were varied-on successfully on the original primary node, clustering calls the exit program on the original primary node by using an action code of Start.

If the exit program for an application CRG is unsuccessful or ends abnormally, cluster resource services attempt to restart the application if the status of the CRG is active. The cluster resource group exit program is called by using an action code of Restart. If the application cannot be restarted in the specified maximum number of attempts, the cluster resource group exit program is called by using an action code of Failover. The restart count is reset only when the exit program is called by using an action code of start, which can be the result of a start CRG, a failover, or a switchover.

When the cluster resource group is started, the application CRG exit program called on the primary node is not to return control to cluster resource services until the application itself ends or an error occurs. After an application CRG is active, if cluster resource services must notify the application CRG exit program of some event, another instance of the exit program is started in a different job. Any action code other than Start or Restart is expected to be returned.

When a cluster resource group exit program is called, it is passed a set of parameters that identify the cluster event being processed, the current state of the cluster resources, and the expected state of the cluster resources.

For complete information about cluster resource group exit programs, including what information is passed to the exit program for each action code, see Cluster Resource Group Exit Program in the Cluster API documentation. Sample source code has been provided in the QUSRTOOL library which can be used as a basis for writing an exit program. See the TCSTAPPEXT member in the QATTSYSC file.

Application CRG considerations

An application cluster resource group manages application resiliency.

Managing application CRG takeover of IP addresses:

You can manage application CRG takeover of IP addresses by using cluster resource services. You can also manage them manually.

You can manage the application takeover IP address that is associated with an application CRG in two ways. The easiest way, which is the default, is to let cluster resource services manage the takeover IP address. This method directs cluster resource services to create the takeover IP address on all nodes in the recovery domain, including nodes subsequently added to the recovery domain. When this method is selected, the takeover IP address cannot currently be defined on any node in the recovery domain.

The alternative way is to manage the takeover IP addresses yourself. This method directs cluster resource services to not take any steps to configure the takeover IP address; the user is responsible for configuration. You must add the takeover IP address on all nodes in the recovery domain (except on replicate nodes) before starting the cluster resource group. Any node to be added to the recovery domain of an active CRG must have the takeover IP address configured before being added.

Related concepts:

"Example: Application cluster resource group failover actions" on page 9
This example shows how one failover scenario works. Other failover scenarios can work differently.

Multiple subnets: It is possible to have the application takeover IP address work across multiple subnets, although the default is to have all recovery domain nodes on the same subnet. To configure the application takeover IP address when the nodes in the recovery domain span multiple subnets, you need to enable the switchover environment.

Enabling application switchover across subnets with IPv4:

Clustering, in general, requires that all cluster nodes in the recovery domain of an application cluster resource group reside on the same LAN (use the same subnet addressing). Cluster resource services supports a user configured takeover IP address when configuring application CRGs.

Address Resolution Protocol (ARP) is the network protocol that is used to switch the configured application takeover IP address from one node to another node in the recovery domain. To enable the application switchover across subnets, you need to use the virtual IP address support and the Routing Information Protocol (RIP) for IPv4.

The following manual configuration steps are required to enable the switchover environment. **This set of instructions must be done on all the nodes in the recovery domain, and repeated for the other nodes in the cluster that will become nodes in the recovery domain for the given application CRG.**

1. Select an IPv4 takeover IP address to be used by the application CRG.
   - To avoid confusion, this address should not overlap with any other existing addresses used by the cluster nodes or routers. For example, if choosing 19.19.19.19, ensure that 19.0.0.0 (19.19.0.0) are not routes known by the system routing tables.
   - Add the takeover interface (for example, 19.19.19.19. Create it with a line description of *VIRTUALIP, subnet mask of 255.255.255.255 (host route), maximum transmission unit of 1500 (any number in the range 576-16388), and autostart of *NO. This takeover address (for example, 19.19.19.19) does must exist as a *VIRTUALIP address before identifying it as an associated local interface in next step. It does not, however, must be active.
2. Associate the intended takeover IP address with one or both of the IP addresses that you specify to be used by cluster communications when you create the cluster or add a node to the cluster.
   - For example, this means that you make the 19.19.19.19 takeover address an associated local interface on the IP address for the cluster node. This must be done for each cluster address on each cluster node.

   Note: The cluster addresses must be ended to accomplish this change under the Configure TCP/IP (CFGTCP) command.
3. Create the cluster and create any CRGs. For the application CRG, specify `QcstUserCfgsTakeoverIpAddr` for the **Configure takeover IP address** field. Do not start any application CRGs.
4. Using Configure TCP/IP applications (option 20) from the Configure TCP/IP menu, then Configure RouteD (option 2), then Change RouteD attributes (option 1), ensure that the Supply field is set to *YES. If not, set it to *YES. Then start or restart RouteD (RIP or RIP-2) on each cluster node.
   - NETSTAT option 3 shows the RouteD using a local port if currently running. RouteD must be running and advertising routes (ensure that the Supply field is set to *YES) on every cluster node in the CRG recovery domain.
5. Ensure that all the commercial routers in the network that interconnect the recovery domain LANs are accepting and advertising host routes for RIP.
   - This is not necessarily the default setting for routers. The language varies with router manufacturer, but the RIP interfaces settings should be set to send host routes and receive dynamic hosts.
   - This also applies to both the router interfaces that point to the systems as well as the router-to-router interfaces.

   Note: Do not use an IBM i machine as the router in this configuration. Use a commercial router (IBM or otherwise) that is designed for routing purposes. IBM i routing cannot be configured to handle this function.
6. Manually activate the takeover address on one of the cluster nodes:
   a. Wait up to 5 minutes for RIP to propagate the routes.

b. Ping the takeover address from all nodes in the CRG recovery domain and from selected clients on the LANs who will be using this address.
c. Ensure the takeover address is ended again.

(Clustering will start the address on the specified primary node when the CRGs are started.)

7. Start the application CRGs.
   - The takeover address is started by clustering on the specified, preferred node, and RIP advertises the routes throughout the recovery domain. RIP might take up to 5 minutes to update routes across the domain. The RIP function is independent from the start CRG function.

Important:

- If the above procedure is not followed for all cluster nodes in the application CRG recovery domain, the cluster hangs during the switchover process.
- Even though you do not perform a failover to replica nodes, it is a good idea to perform the procedure on the replica nodes in the event that they might be changed at a later date in time to become a backup.
- If you want to use multiple virtual IP addresses, then each one will require a separate application CRG and a separate IP address with which to be associated. This address may be another logical IP address on the same physical adapter or it may be another physical adapter altogether. Also, care must be taken to prevent ambiguities in the routing tables. This is best achieved by doing the following:
  - Add a *DFTROUTE to the routing table for each virtual IP address.
  - To use multiple IP address use CFGTCP (option 2).
  - Set all parameters, including the next hop, the same to reach the router of choice; however, the Preferred binding interface should be set to the local system IP address that is associated with the virtual IP address that is represented by this route.

Enabling application switchover across subnets with IPv6:

Clustering, in general, requires that all cluster nodes in the recovery domain of an application cluster resource group reside on the same LAN (use the same subnet addressing). Cluster resource services supports a user configured takeover IP address when configuring application CRGs.

Address Resolution Protocol (ARP) is the network protocol that is used to switch the configured application takeover IP address from one node to another node in the recovery domain. To enable the application switchover across subnets, you need to use the virtual IP address support and the Routing Information Protocol Next Generation (RIPng) for IPv6.

The following manual configuration steps are required to enable the switchover environment. **This set of instructions must be done on all the nodes in the recovery domain, and repeated for the other nodes in the cluster that will become nodes in the recovery domain for the given application CRG.**

1. Select an IPv6 takeover IP address to be used by the application CRG.
   - To avoid confusion, this address should not overlap with any other existing addresses used by the cluster nodes or routers.
   - It is recommended that this address is defined with a shorter IPv6 address prefix than any other IPv6 address that shares the same IPv6 prefix to ensure that the correct address is chosen for the source address in outbound packets.
   - Add the takeover interface (for example, 2001:0DB8:1234::1. Create it with a line description of *VIRTUALIP, maximum transmission unit of 1500 (any number in the range 576-16388), and autostart of *NO.
2. Create the cluster and create any CRGs. For the application CRG, specify `QcstUserCfgsTakeoverIpAddr` for the **Configure takeover IP address** field. Do not start any application CRGs.

3. Use the Change RIP Attributes (CHGRIPA) command to set the RIPng attributes. Run the command: CHGRIPA AUTOSTART(*YES) IP6COND(*NEVER) IP6ACPDFT(*NO) IP6SNDONLY(*VIRTUAL).
4. Ensure there is an IPv6 link-local address active on the system. An IPv6 link-local address starts with 'fe80:'.
5. Use the Add RIP Interface (ADDRIPIFC) command add a RIP interface used by the OMPROUTED server to advertise the virtual address used for the takeover IP address. For example, if fe80::1 is the active IPv6 link-local address, run the command: ADDRIPIFC IFC('fe80::1') RCVDYNNET(*YES) SNDSTTRTE(*YES) SNDHOSTRTE(*YES) SNDONLY(*VIRTUAL).
6. Restart the OMPROUTED server using the following commands:
   a. ENDTCPSVR SERVER(*OMPROUTED) INSTANCE(*RIP)
   b. STRTCPSVR SERVER(*OMPROUTED) INSTANCE(*RIP)
7. Ensure that all the commercial routers in the network that interconnect the recovery domain LANs are accepting and advertising host routes for RIPng.
   - This is not necessarily the default setting for routers. The language varies with router manufacturer, but the RIPng interfaces settings should be set to send host routes and receive dynamic hosts.
   - This also applies to both the router interfaces that point to the systems as well as the router-to-router interfaces.

   Note: Do not use an IBM i machine as the router in this configuration. Use a commercial router (IBM or otherwise) that is designed for routing purposes. IBM i routing cannot be configured to handle this function.
8. Manually activate the takeover address on one of the cluster nodes:
   a. Wait up to 5 minutes for RIP to propagate the routes.
   b. Ping the takeover address from all nodes in the CRG recovery domain and from selected clients on the LANs who will be using this address.
   c. Ensure that the takeover address is ended again.

   (Clustering will start the address on the specified primary node when the CRGs are started.)
9. Start the application CRGs.
   - The takeover address is started by clustering on the specified, preferred node, and RIPng advertises the routes throughout the recovery domain. RIPng might take up to 5 minutes to update routes across the domain. The RIPng function is independent from the start CRG function.

Important:

- If the above procedure is not followed for all cluster nodes in the application CRG recovery domain, the cluster hangs during the switchover process.
- Even though you do not perform a failover to replica nodes, it is a good idea to perform the procedure on the replica nodes in the event that they might be changed at a later date in time to become a backup.
- If you want to use multiple virtual IP addresses, then each one will require a separate application CRG and a separate IP address with which to be associated. This address may be another logical IP address on the same physical adapter or it may be another physical adapter altogether. Also, care must be taken to prevent ambiguities in the routing tables. This is best achieved by doing the following:
  - Add a *DFTROUTE to the routing table for each virtual IP address.
  - To use multiple IP address use CFGTCP (option 2).
  - Set all parameters, including the next hop, the same to reach the router of choice; however, the Preferred binding interface should be set to the local system IP address that is associated with the virtual IP address that is represented by this route.

Example: Application cluster resource group failover actions

This example shows how one failover scenario works. Other failover scenarios can work differently.

The following happens when a cluster resource group for a resilient application fails over due to exceeding the retry limit or if the job is canceled:

- The cluster resource group exit program is called on all active nodes in the recovery domain for the CRG with an action code of failover. This indicates that cluster resource services is preparing to failover the application's point of access to the first backup.
- Cluster resource services ends the takeover Internet Protocol (IP) connection on the primary node.
- Cluster resource services starts the takeover IP address on the first backup (new primary) node.
- Cluster resource services submits a job that calls the cluster resource group exit program only on the new primary node with an action code of Start. This action restarts the application.

Related concepts:

"Managing application CRG takeover of IP addresses" on page 6
You can manage application CRG takeover of IP addresses by using cluster resource services. You can also manage them manually.

Example: Application exit program

This code example contains an application cluster resource group exit program.

You can find this code example in the QUSRTOOL library.

Note: By using the code examples, you agree to the terms of the "Code license and disclaimer information" on page 196.

```
/*****************************************************************************/
/*                                                                           */
/* Library:  QUSRTOOL                                                        */
/* File:     QATTSYSC                                                        */
/* Member:   TCSTAPPEXT                                                      */
/* Type:     ILE C                                                           */
/*                                                                           */
/* Description:                                                              */
/* This is an example application CRG exit program which gets called for    */
/* various cluster events or cluster APIs.  The bulk of the logic must      */
/* still be added because that logic is really dependent upon the unique    */
/* things that need to be done for a particular application.                */
/*                                                                           */
/* The intent of this example to to provide a shell which contains the      */
/* basics for building a CRG exit program.  Comments throughout the example*/
/* highlight the kinds of issues that need to be addressed by the real      */
/* exit program implementation.                                              */
/*                                                                           */
/* Every action code that applies to an application CRG is handled in this */
/* example.                                                                  */
/*                                                                           */
/* The tcstdtaara.h include is also shipped in the QUSRTOOL library.  See   */
/* the TCSTDTAARA member in the QATTSYSC file.                               */
/*                                                                           */
/* Change log:                                                               */
/* Flag Reason   Ver    Date   User Id   Description                         */
/* ____ ________ ______ ______ ________ ____________________________________ */
/* ...  D98332   v5r1m0 000509 ROCH     Initial creation.                    */
/* $A1  P9950070 v5r2m0 010710 ROCH     Dataarea fixes                       */
/* $A2  D99055   v5r2m0 010913 ROCH     Added CancelFailover action code     */
/* $A3  D98854   v5r2m0 010913 ROCH     Added VerificationPhase action code*/
/* $A4  P9A10488 v5r3m0 020524 ROCH     Added example code to wait for data*/
/*                                      CRGs on switchover action code       */
/*                                                                           */
/*****************************************************************************/


/*---------------------------------------------------------------------------*/
/*                                                                           */
/* Header files                                                              */
```

```
/*                                                                    */
/*--------------------------------------------------------------------*/
#include              /* Useful when debugging                   */
#include             /* offsetof macro                           */
#include             /* system function                          */
#include             /* String functions                         */
#include             /* Exception handling constants/structures  */
#include               /* Various cluster constants               */
#include           /* Structure of CRG information               */
#include "qusrtool/qattsysc/tcstdtaara" /* QCSTHAAPPI/QCSTHAAPPO data areas*/
#include           /* API to Retrieve contents of a data area     */
#include              /* API error code type definition           */
#include              /* mitime builtin                           */
#include           /* waittime builtin                           */


/*--------------------------------------------------------------------*/
/*                                                                    */
/* Constants                                                          */
/*                                                                    */
/*--------------------------------------------------------------------*/
#define UnknownRole -999
#define DependCrgDataArea "QCSTHAAPPO"
#define ApplCrgDataArea "QCSTHAAPPI"
#define Nulls 0x0000000000000000000


/*--------------------------------------------------------------------*/
/*                                                                    */
/* The following constants are used in the checkDependCrgDataArea()   */
/* function.  The first defines how long to sleep before checking the data */
/* area.  The second defines that maximum time to wait for the data area   */
/* to become ready before failing to start the application when the Start  */
/* CRG function is being run.  The third defines the maximum wait time for */
/* the Initiate Switchover or failover functions.                     */
/*                                                                    */
/*--------------------------------------------------------------------*/
#define WaitSecondsIncrement 30
#define MaxStartCrgWaitSeconds 0
#define MaxWaitSeconds 900

/*--------------------------------------------------------------------*/
/*                                                                    */
/* As this exit program is updated to handle new action codes, change the  */
/* define below to the value of the highest numbered action code that is   */
/* handled.                                                           */
/*                                                                    */
/*--------------------------------------------------------------------*/
#define MaxAc 21

/*--------------------------------------------------------------------*/
/*                                                                    */
/* If the exit program data in the CRG has a particular structure to it,   */
/* include the header file for that structure definition and change the    */
/* define below to use that structure name rather than char.          */
/*                                                                    */
/*--------------------------------------------------------------------*/
#define EpData char

/*--------------------------------------------------------------------*/
/*                                                                    */
/* Change the following define to the library the application resides in   */
/* and thus where the QCSTHAAPPO and QCSTHAAPPI data areas will be found.  */
/*                                                                    */
/*--------------------------------------------------------------------*/
#define ApplLib "QGPL"
```

```
/*--------------------------------------------------------------------------*/
/*                                                                          */
/* Prototypes for internal functions.                                       */
/*                                                                          */
/*--------------------------------------------------------------------------*/
static int getMyRole(Qcst_EXTP0100_t *, int, int);
#pragma argopt(getMyRole)
static int doAction(int, int, int, Qcst_EXTP0100_t *, EpData *);
#pragma argopt(doAction)
static int createCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int startCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int restartCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int endCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int verifyPhase(int, int, Qcst_EXTP0100_t *, EpData *);
static int deleteCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int memberIsJoining(int, int, Qcst_EXTP0100_t *, EpData *);
static int memberIsLeaving(int, int, Qcst_EXTP0100_t *, EpData *);
static int switchPrimary(int, int, Qcst_EXTP0100_t *, EpData *);
static int addNode(int, int, Qcst_EXTP0100_t *, EpData *);
static int rmvNode(int, int, Qcst_EXTP0100_t *, EpData *);
static int chgCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int deleteCrgWithCmd(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoPriorAction(int, int, Qcst_EXTP0100_t *, EpData *);
static int endNode(int, int, Qcst_EXTP0100_t *, EpData *);
static int chgNodeStatus(int, int, Qcst_EXTP0100_t *, EpData *);
static int cancelFailover(int, int, Qcst_EXTP0100_t *, EpData *);
static int newActionCode(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoCreateCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoStartCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoEndCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoMemberIsJoining(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoMemberIsLeaving(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoSwitchPrimary(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoAddNode(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoRmvNode(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoChgCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoCancelFailover(int, int, Qcst_EXTP0100_t *, EpData *);
static void bldDataAreaName(char *, char *, char *);
#pragma argopt(bldDataAreaName)
static int checkDependCrgDataArea(unsigned int);
#pragma argopt(checkDependCrgDataArea)
static void setApplCrgDataArea(char);
#pragma argopt(setApplCrgDataArea)
static void cancelHandler(_CNL_Hndlr_Parms_T *);
static void unexpectedExceptionHandler(_INTRPT_Hndlr_Parms_T *);
static void endApplication(unsigned int, int, int, Qcst_EXTP0100_t *, EpData *);
#pragma argopt(endApplication)

/*--------------------------------------------------------------------------*/
/*                                                                          */
/* Some debug routines                                                      */
/*                                                                          */
/*--------------------------------------------------------------------------*/
static void printParms(int, int, int, Qcst_EXTP0100_t *, EpData *);
static void printActionCode(unsigned int);
static void printCrgStatus(int);
static void printRcvyDomain(char *,
                            unsigned int,
                            Qcst_Rcvy_Domain_Array1_t *);
static void printStr(char *, char *, unsigned int);


/*--------------------------------------------------------------------------*/
/*                                                                          */
/* Type definitions                                                         */
/*                                                                          */
```

```
/*--------------------------------------------------------------------*/

/*--------------------------------------------------------------------*/
/*                                                                    */
/* This structure defines data that will be passed to the exception and    */
/* cancel handlers.  Extend it with information unique to your application.*/
/*                                                                    */
/*--------------------------------------------------------------------*/
typedef struct {
  int *retCode;             /* Pointer to return code                 */
  EpData *epData;           /* Exit program data from the CRG         */
  Qcst_EXTP0100_t *crgData; /* CRG data                               */
  unsigned int actionCode;  /* The action code                        */
  int role;                 /* This node's recovery domain role       */
  int priorRole;            /* This node's prior recovery domainrole  */
} volatile HandlerDataT;


/*--------------------------------------------------------------------*/
/*                                                                    */
/* Function pointer array for handling action codes.  When the exit program*/
/* is updated to handle new action codes, add the new function names to    */
/* this function pointer array.                                       */
/*                                                                    */
/*--------------------------------------------------------------------*/
static int (*fcn[MaxAc+1]) (int role,
                            int priorRole,
                            Qcst_EXTP0100_t *crgData,
                            EpData *epData) = {
  newActionCode,   /* 0 - currently reserved */
  createCrg,       /* 1  */
  startCrg,        /* 2  */
  restartCrg,      /* 3  */
  endCrg,          /* 4  */
  verifyPhase,     /* 5 - currently reserved */
  newActionCode,   /* 6 - currently reserved */
  deleteCrg,       /* 7  */
  memberIsJoining, /* 8  */
  memberIsLeaving, /* 9  */
  switchPrimary,   /* 10 */
  addNode,         /* 11 */
  rmvNode,         /* 12 */
  chgCrg,          /* 13 */
  deleteCrgWithCmd,/* 14 */
  undoPriorAction, /* 15 */
  endNode,         /* 16 */
  newActionCode,   /* 17 - applies only to a device CRG */
  newActionCode,   /* 18 - applies only to a device CRG */
  newActionCode,   /* 19 - applies only to a device CRG */
  chgNodeStatus,   /* 20 */
  cancelFailover   /* 21 */
};


/*--------------------------------------------------------------------*/
/*                                                                    */
/* Function pointer array for handling prior action codes when called with */
/* the Undo action code.  When the exit program is updated to handle       */
/* Undo for new action codes, add the new function names to this function  */
/* pointer array.                                                     */
/*                                                                    */
/*--------------------------------------------------------------------*/
static int (*undoFcn[MaxAc+1]) (int role,
                                int priorRole,
                                Qcst_EXTP0100_t *crgData,
                                EpData *epData) = {
  newActionCode,       /* 0 - currently reserved */
```

```
  undoCreateCrg,       /* 1  */
  undoStartCrg,        /* 2  */
  newActionCode,       /* 3  */
  undoEndCrg,          /* 4  */
  newActionCode,       /* 5 - no undo for this action code */
  newActionCode,       /* 6 - currently reserved */
  newActionCode,       /* 7  */
  undoMemberIsJoining, /* 8  */
  undoMemberIsLeaving, /* 9  */
  undoSwitchPrimary,   /* 10 */
  undoAddNode,         /* 11 */
  undoRmvNode,         /* 12 */
  undoChgCrg,          /* 13 */
  newActionCode,       /* 14 */
  newActionCode,       /* 15 */
  newActionCode,       /* 16 */
  newActionCode,       /* 17 - applies only to a device CRG */
  newActionCode,       /* 18 - applies only to a device CRG */
  newActionCode,       /* 19 - applies only to a device CRG */
  newActionCode,       /* 20 */
  undoCancelFailover   /* 21 */
};



/*****************************************************************************/
/*                                                                           */
/* This is the entry point for the exit program.                             */
/*                                                                           */
/*****************************************************************************/
void main(int argc, char *argv[]) {

  HandlerDataT hdlData;


/*--------------------------------------------------------------------------- */
  /*                                                                         */
  /* Take each of the arguments passed in the argv array and castit to       */
  /* the correct data type.                                                  */
  /*                                                                         */

/*--------------------------------------------------------------------------- */
  int *retCode    = (int *)argv[1];
  unsigned int *actionCode = (unsigned int *)argv[2];
  EpData *epData           = (EpData *)argv[3];
  Qcst_EXTP0100_t *crgData = (Qcst_EXTP0100_t *)argv[4];
  char *formatName         = (char *)argv[5];


/*---------------------------------------------------------------------------*/
  /*                                                                         */
  /* Ensure the format of the data being passed is correct.  */
  /* If not, a change has been made and this exit program needs tobe      */
  /* updated to accomodate the change.  Add appropriate errorlogging for  */
  /* your application design.                                                */
  /*                                                                         */

/*---------------------------------------------------------------------------*/
  if (0 != memcmp(formatName, "EXTP0100", 8))
    abort();


/*---------------------------------------------------------------------------*/
  /*                                                                         */
  /* Set up the data that will be passed to the exception andcancel       */
  /* handlers.                                                               */
  /*                                                                         */
```

```
/*------------------------------------------------------------------------*/
  hdlData.retCode    = retCode;
  hdlData.epData     = epData;
  hdlData.crgData    = crgData;
  hdlData.actionCode = *actionCode;
  hdlData.role       = UnknownRole;
  hdlData.priorRole  = UnknownRole;
  _VBDY();  /* force changed variables to home storage location          */


/*------------------------------------------------------------------------*/
  /*                                                                      */
  /* Enable an exception handler for any and all exceptions.              */
  /*                                                                      */

/*------------------------------------------------------------------------*/
#pragma exception_handler(unexpectedExceptionHandler, hdlData, \
                          _C1_ALL, _C2_ALL, _CTLA_INVOKE )


/*------------------------------------------------------------------------*/
  /*                                                                      */
  /* Enable a cancel handler to recover if this job is canceled.         */
  /*                                                                      */

/*------------------------------------------------------------------------*/
#pragma cancel_handler(cancelHandler, hdlData)


/*------------------------------------------------------------------------*/
  /*                                                                      */
  /* Extract the role and prior role of the node this exit program is     */
  /* running on.  If the cluster API or event changes the recovery domain */
  /* (node role or membership status), the new recovery domain's offset is */
  /* passed in Offset_Rcvy_Domain_Array and the offset of the recovery    */
  /* domain as it looked prior to the API or cluster event is passed in   */
  /* Offset_Prior_Rcvy_Domain_Array.  If the recovery domain isn't changed,*/
  /* only Offset_Rcvy_Domain_Array can be used to address the recovery    */
  /* domain.                                                              */
  /*                                                                      */

/*------------------------------------------------------------------------*/
  hdlData.role = getMyRole(crgData,
                           crgData->Offset_Rcvy_Domain_Array,
                           crgData->Number_Nodes_Rcvy_Domain);
  if (crgData->Offset_Prior_Rcvy_Domain_Array)
    hdlData.priorRole =
                getMyRole(crgData,

crgData->Offset_Prior_Rcvy_Domain_Array,

crgData->Number_Nodes_Prior_Rcvy_Domain);
  else
    hdlData.priorRole = hdlData.role;
  _VBDY();  /* force changed variables to home storage location          */


/*------------------------------------------------------------------------*/
  /*                                                                      */
  /* Enable the following to print out debug information.                 */
  /*                                                                      */

/*------------------------------------------------------------------------*/
  /*
  printParms(*actionCode, hdlData.role, hdlData.priorRole, crgData,
epData);
```

```
  */


/*--------------------------------------------------------------------*/
  /*                                                                  */
  /* Do the correct thing based upon the action code.  The return code*/
  /* is set to the function result of doAction().                     */
  /*                                                                  */

/*--------------------------------------------------------------------*/
  *retCode = doAction(*actionCode,
                      hdlData.role,
                      hdlData.priorRole,
                      crgData,
                      epData);


/*--------------------------------------------------------------------*/
  /*                                                                  */
  /* The exit program job will end when control returns to the operating*/
  /* system at this point.                                            */
  /*                                                                  */

/*--------------------------------------------------------------------*/
  return;

#pragma disable_handler  /* unexpectedExceptionHandler                */
#pragma disable_handler  /* cancelHandler                             */
}  /* end main()                                                      */


/**********************************************************************/
/*                                                                    */
/* Get the role of this particular node from one of the views of the  */
/* recovery domain.                                                   */
/*                                                                    */
/* APIs and cluster events which pass the updated and prior recovery domain*/
/* to the exit program are:                                           */
/* QcstAddNodeToRcvyDomain                                            */
/* QcstChangeClusterNodeEntry                                         */
/* QcstChangeClusterResourceGroup                                     */
/* QcstEndClusterNode (ending node does not get the prior domain)     */
/* QcstInitiateSwitchOver                                             */
/* QcstRemoveClusterNodeEntry (removed node does not get the prior domain) */
/* QcstRemoveNodeFromRcvyDomain                                       */
/* QcstStartClusterResourceGroup (only if inactive backup nodes are   */
/*                                reordered)                          */
/* a failure causing failover                                         */
/* a node rejoining the cluster                                       */
/* cluster partitions merging                                         */
/*                                                                    */
/* All other APIs pass only the updated recovery domain.              */
/*                                                                    */
/**********************************************************************/
static int getMyRole(Qcst_EXTP0100_t *crgData, int offset, int
count) {

  Qcst_Rcvy_Domain_Array1_t *nodeData;
  unsigned int iter = 0;


/*--------------------------------------------------------------------*/
  /*                                                                  */
  /* Under some circumstances, the operating system may not be able to*/
  /* determine the ID of this node and passes *NONE.  An example of such a */
  /* circumstance is when cluster resource services is not active on a*/
  /* node and the DLTCRG CL command is used.                          */
```

```
  /*                                                                  */

/*-------------------------------------------------------------------*/
  if (0 == memcmp(crgData->This_Nodes_ID, QcstNone,
sizeof(Qcst_Node_Id_t)))
    return UnknownRole;


/*-------------------------------------------------------------------*/
  /*                                                                  */
  /* Compute a pointer to the first element of the recovery domain array.  */
  /*                                                                  */

/*-------------------------------------------------------------------*/
  nodeData = (Qcst_Rcvy_Domain_Array1_t *)((char *)crgData +
offset);


/*-------------------------------------------------------------------*/
  /*                                                                  */
  /* Find my node in the recovery domain array.  I will not be in the      */
  /* prior recovery domain if I am being added by the Add Node to Recovery */
  /* Domain API.                                                      */
  /*                                                                  */

/*-------------------------------------------------------------------*/
  while (  0 != memcmp(crgData->This_Nodes_ID,
                       nodeData->Node_ID,
                       sizeof(Qcst_Node_Id_t))
         &&
           iter &lt; count
        ) {
    nodeData++;
    iter++;
  }

  if (iter &lt; count)
    return nodeData->Node_Role;
  else
    return UnknownRole;
}  /* end getMyRole()                                               */


/*********************************************************************/
/*                                                                   */
/* Call the correct function based upon the cluster action code.  The      */
/* doAction() function was split out from main() in order to clarify the   */
/* example.  See the function prologues for each called function for       */
/* information about a particular cluster action.                    */
/*                                                                   */
/* Each action code is split out into a separate function only to help     */
/* clarify this example.  For a particular exit program, some action codes */
/* may perform the same function in which case multiple action codes could */
/* be handled by the same function.                                  */
/*                                                                   */
/*********************************************************************/
static int doAction(int actionCode,
                    int role,
                    int priorRole,
                    Qcst_EXTP0100_t *crgData,
                    EpData *epData) {


/*-------------------------------------------------------------------*/
  /*                                                                  */
  /* For action codes this exit program knows about, call a function to    */
  /* do the work for that action code.                                */
```

```
  /*                                                                  */

/*------------------------------------------------------------------*/

  if (actionCode &lt;= MaxAc )
    return (*fcn[actionCode]) (role, priorRole, crgData, epData);
  else

/*------------------------------------------------------------------*/
    /*                                                              */
    /* IBM has defined a new action code in a new operating system release */
    /* and this exit program has not yet been updated to handle it.  Take a*/
    /* default action for now.                                      */
    /*                                                              */

/*------------------------------------------------------------------*/
    return newActionCode(role, priorRole, crgData, epData);
}  /* end doAction()                                                */


/********************************************************************/
/*                                                                  */
/* Action code = QcstCrgAcInitialize                                */
/*                                                                  */
/* The QcstCreateClusterResourceGroup API was called.  A new cluster      */
/* resource group object is being created.                          */
/*                                                                  */
/* Things to consider:                                              */
/*   - Check that the application program and all associated objects are on*/
/*     the primary and backup nodes.  If the objects are not there,        */
/*     consider sending error/warning messages or return a failure return  */
/*     code.                                                        */
/*   - Check that required data or device CRGs are on all nodes in the     */
/*     recovery domain.                                             */
/*   - Perform any necessary setup that is required to run the             */
/*     the application on the primary or backup nodes.              */
/*   - If this CRG is enabled to use the QcstDistributeInformation API,    */
/*     the user queue needed by that API could be created at this time.    */
/*                                                                  */
/********************************************************************/
static int createCrg(int role,
                     int doesNotApply,
                     Qcst_EXTP0100_t *crgData,
                     EpData *epData) {

  return QcstSuccessful;
}  /* end createCrg()                                               */


/********************************************************************/
/*                                                                  */
/* Action code = QcstCrgAcStart                                     */
/*                                                                  */
/* The QcstStartClusterResourceGroup API was called.  A cluster resource   */
/* group is being started.                                          */
/* The QcstInitiateSwitchOver API was called and this is the second action */
/* code being passed to the exit program.                           */
/* The fail over event occurred and this is the second action code being   */
/* passed to the exit program.                                      */
/*                                                                  */
/* A maximum wait time is used when checking to see if all dependent CRGs  */
/* are active.  This is a short time if the CRG is being started because of*/
/* the QcstStartClusterResourceGroup API.  It is a longer time if it is    */
/* because of a failover or switchover.  When failover or switchover are   */
/* being done, it make take a while for data or device CRGs to become      */
/* ready so the wait time is long.  If the Start CRG API is being used, the*/
/* dependent CRGs should already be started or some error occurred, the    */
```

```
/* CRGs were started out of order, etc. and there is no need for a long    */
/* wait.                                                                   */
/*                                                                         */
/* Things to consider:                                                     */
/*   - If this node's role is primary, the application should be started.  */
/*     This exit program should either call the application so that it runs*/
/*     in this same job or it should monitor any job started by this       */
/*     exit program so the exit program knows when the application job     */
/*     ends.  By far, the simplest approach is run the application in this */
/*     job by calling it.                                                  */
/*     Cluster Resource Services is not expecting this exit program to     */
/*     return until the application finishes running.                      */
/*   - If necessary, start any associated subsystems, server jobs, etc.    */
/*   - Ensure that required data CRGs have a status of active on all nodes */
/*     in the recovery domain.                                             */
/*                                                                         */
/*************************************************************************/
static int startCrg(int role,
                    int doesNotApply,
                    Qcst_EXTP0100_t *crgData,
                    EpData *epData) {

  unsigned int maxWaitTime;

  /* Start the application if this node is the primary                     */
  if (role == QcstPrimaryNodeRole) {

/*-----------------------------------------------------------------------*/
    /*                                                                     */
    /* Determine if all CRGs that this application CRG is dependent upon   */
    /* are ready.  If the check fails, return from the Start action code.  */
    /* Cluster Resource Services will change the state of the CRG to       */
    /* Inactive.                                                           */
    /*                                                                     */

/*-----------------------------------------------------------------------*/
    if (crgData->Cluster_Resource_Group_Status ==
QcstCrgStartCrgPending)
      maxWaitTime = MaxStartCrgWaitSeconds;
    else
      maxWaitTime = MaxWaitSeconds;
    if (QcstSuccessful != checkDependCrgDataArea(maxWaitTime))
      return QcstSuccessful;



/*-----------------------------------------------------------------------*/
    /*                                                                     */
    /* Just before starting the application, update the data area to       */
    /* indicate the application is running.                                */
    /*                                                                     */

/*-----------------------------------------------------------------------*/
    setApplCrgDataArea(Appl_Running);


/*-----------------------------------------------------------------------*/
    /*                                                                     */
    /* Add logic to call application here.  It is expected that control    */
    /* will not return until something causes the application to end: a    */
    /* normal return from the exit program, the job is canceled, or an    */
    /* unhandled exception occurs.  See the cancelHandler() function for   */
    /* some common ways this job could be canceled.                        */
    /*                                                                     */

/*-----------------------------------------------------------------------*/
```

```
/*--------------------------------------------------------------------*/
    /*                                                                */
    /* After the application has ended normally, update the data area to   */
    /* indicate the application is no longer running.                 */
    /*                                                                */

/*--------------------------------------------------------------------*/
    setApplCrgDataArea(Appl_Ended);
  }
  else

/*--------------------------------------------------------------------*/
    /*                                                                */
    /* On backup or replicate nodes, mark the status of the application in */
    /* the data area as not running.                                  */
    /*                                                                */

/*--------------------------------------------------------------------*/
    setApplCrgDataArea(Appl_Ended);


  return QcstSuccessful;
}  /* end startCrg()
       */


/**********************************************************************/
/*                                                                    */
/* Action code = QcstCrgAcRestart                                     */
/*                                                                    */
/* The previous call of the exit program failed and set the return   */
/* code to QcstFailWithRestart or it failed due to an exception and the    */
/* exception was allowed to percolate up the call stack.  In either  */
/* case, the maximum number of times for restarting the exit program has   */
/* not been reached yet.                                              */
/*                                                                    */
/* This action code is passed only to application CRG exit programs which  */
/* had been called with the Start action code.                        */
/*                                                                    */
/**********************************************************************/
static int restartCrg(int role,
                      int doesNotApply,
                      Qcst_EXTP0100_t *crgData,
                      EpData *epData) {


/*--------------------------------------------------------------------*/
  /*                                                                */
  /* Perform any unique logic that may be necessary when restarting the    */
  /* application after a failure and then call the startCrg() function to  */
  /* do the start functions.                                        */
  /*                                                                */

/*--------------------------------------------------------------------*/


  return startCrg(role, doesNotApply, crgData, epData);
}  /* end restartCrg()                                                */


/**********************************************************************/
/*                                                                    */
/* Action code = QcstCrgAcEnd                                         */
/*                                                                    */
```

```
/* The end action code is used for one of the following reasons:          */
/*   - The QcstEndClusterResourceGroup API was called.                    */
/*   - The cluster has become partitioned and this node is in the secondary*/
/*     partition.  The End action code is used regardless of whether the  */
/*     CRG was active or inactive.  Action code dependent data of         */
/*     QcstPartitionFailure will also be passed.                          */
/*   - The application ended.   Action code dependent data of             */
/*     QcstResourceEnd will also be passed.  All nodes in the recovery    */
/*     domain will see the same action code (including the primary).      */
/*   - The CRG job has been canceled.  The exit program on this node will */
/*     be called with the End action code.  QcstMemberFailure will be     */
/*     passed as action code dependent data.                              */
/*                                                                        */
/*                                                                        */
/*                                                                        */
/* Things to consider:                                                    */
/*   - If the CRG is active, the job running the application is canceled  */
/*     and the IP takeover address is ended AFTER the exit program is     */
/*     called.                                                            */
/*   - If subsystems or server jobs were started as a result of the       */
/*     QcstCrgAcStart action code, end them here or consolidate all logic */
/*     to end the application in the cancelHandler() since it will be     */
/*     invoked for all Cluster Resource Services APIs which must end the  */
/*     application on the current primary.                                */
/*                                                                        */
/**************************************************************************/
static int endCrg(int role,
                  int priorRole,
                  Qcst_EXTP0100_t *crgData,
                  EpData *epData) {


/*------------------------------------------------------------------------*/
  /*                                                                      */
  /* End the application if it is running on this node.                   */
  /*                                                                      */

/*------------------------------------------------------------------------*/
  endApplication(QcstCrgAcRemoveNode, role, priorRole, crgData,
epData);

  return QcstSuccessful;
}  /* end endCrg()                                                       */


/**************************************************************************/
/*                                                                        */
/* Action code = QcstCrgAcVerificationPhase                               */
/*                                                                        */
/* The verification phase action code is used to allow the exit program to */
/* do some verification before proceeding with the requested function     */
/* identified by the action code depended data. If the exit program       */
/* determines that the requested function cannot proceed it should return */
/* QcstFailWithOutRestart.                                                */
/*                                                                        */
/*                                                                        */
/* NOTE: The exit program will NOT be called with Undo action code.       */
/*                                                                        */
/**************************************************************************/
static int verifyPhase(int role,
                       int doesNotApply,
                       Qcst_EXTP0100_t *crgData,
                       EpData *epData) {


/*------------------------------------------------------------------------*/
  /*                                                                      */
```

```
  /* Do verification                                                    */
  /*                                                                    */

/*--------------------------------------------------------------------*/
  if (crgData->Action_Code_Dependent_Data == QcstDltCrg) {
        /* do verification  */
        /* if ( fail ) */
          /* return QcstFailWithOutRestart */
  }

  return QcstSuccessful;
}  /* end verifyPhase()                                               */


/**********************************************************************/
/*                                                                    */
/* Action code = QcstCrgAcDelete                                      */
/*                                                                    */
/* The QcstDeleteClusterResourceGroup or QcstDeleteCluster API was called. */
/* A cluster resource group is being deleted while Cluster Resource   */
/* Services is active.                                                */
/* If the QcstDeleteCluster API was used, action code dependent data of  */
/* QcstDltCluster is passed.                                          */
/* If the QcstDeleteCluster API was used and the CRG is active, the exit  */
/* program job which is still active for the Start action code is canceled*/
/* after the Delete action code is processed.                         */
/*                                                                    */
/* Things to consider:                                                */
/*   - Delete application programs and objects from nodes where they are  */
/*     no longer needed such as backup nodes.  Care needs to be exercised  */
/*     when deleting application objects just because a CRG is being  */
/*     deleted since a particular scenario may want to leave the      */
/*     application objects on all nodes.                              */
/*                                                                    */
/**********************************************************************/
static int deleteCrg(int role,
                     int doesNotApply,
                     Qcst_EXTP0100_t *crgData,
                     EpData *epData) {

  return QcstSuccessful;
}  /* end deleteCrg()
      */


/**********************************************************************/
/*                                                                    */
/* Action code = QcstCrgAcReJoin                                      */
/*                                                                    */
/* One of three things is occurring-                                  */
/*  1. The problem which caused the cluster to become partitioned has been */
/*     corrected and the 2 partitions are merging back together to become  */
/*     a single cluster.  Action code dependent data of QcstMerge will be  */
/*     passed.                                                        */
/*  2. A node which either previously failed or which was ended has had   */
/*     cluster resource services started again and the node is joining the */
/*     cluster.  Action code dependent data of QcstJoin will be passed.   */
/*  3. The CRG job on a particular node which may have been canceled or  */
/*     ended has been restarted.  Action code dependent data of QcstJoin  */
/*     will be passed.                                                */
/*                                                                    */
/* Things to consider:                                                */
/*   - If the application replicates application state information to other*/
/*     nodes when the application is running, this state information will  */
/*     need to be resynchronized with the joining nodes if the CRG is     */
/*     active.                                                        */
/*   - Check for missing application objects on the joining nodes.        */
```

```
/*   - Ensure the required data CRGs are on the joining nodes.            */
/*   - If the application CRG is active, ensure the required data CRGs are */
/*     active.                                                            */
/*                                                                        */
/**************************************************************************/
static int memberIsJoining(int role,
                           int priorRole,
                           Qcst_EXTP0100_t *crgData,
                           EpData *epData) {


/*------------------------------------------------------------------------*/
  /*                                                                      */
  /* Ensure the data area status on this node starts out indicating      */
  /* the application is not running if this node is not the primary.     */
  /*                                                                      */

/*------------------------------------------------------------------------*/
  if (role != QcstPrimaryNodeRole) {
    setApplCrgDataArea(Appl_Ended);
  }


/*------------------------------------------------------------------------*/
  /*                                                                      */
  /* If a single node is rejoining the cluster, you may do a certain set of*/
  /* actions.  Whereas if the nodes in a cluster which became partitioned  */
  /* are merging back together, you may have a different set of actions.   */
  /*                                                                      */

/*------------------------------------------------------------------------*/
  if (crgData->Action_Code_Dependent_Data == QcstJoin) {
    /* Do actions for a node joining.                                     */
  }
  else {
    /* Do actions for partitions merging.                                 */
  }

  return QcstSuccessful;
}  /* end memberIsJoining()                                               */


/**************************************************************************/
/*                                                                        */
/* Action code = QcstCrgAcFailover                                        */
/*                                                                        */
/* Cluster resource services on a particular node(s) has failed or ended  */
/* for this cluster resource group.  The Failover action code is passed   */
/* regardless of whether the CRG is active or inactive.  Failover can     */
/* happen for a number of reasons:                                        */
/*                                                                        */
/*   - an operator canceled the CRG job on a node.  Action code dependent */
/*     data of QcstMemberFailure will be passed.                          */
/*   - cluster resource services was ended on the node (for example, the  */
/*     QSYSWRK subsystem was ended with CRS still active).  Action code   */
/*     dependent data of QcstNodeFailure will be passed.                  */
/*   - the application for an application CRG has failed on the primary   */
/*     node and could not be restarted there.  The CRG is Active.         */
/*     Action code dependent data of QcstApplFailure will be passed.      */
/*   - the node failed (such as a power failure).  Action code dependent  */
/*     data of QcstNodeFailure will be passed.                            */
/*   - The cluster has become partitioned due to some communication failure*/
/*     such as a communication line or LAN failure.  The Failover action  */
/*     code is passed to recovery domain nodes in the majority partition. */
/*     Nodes in the minority partition see the End action code.  Action   */
/*     code dependent data of QcstPartitionFailure will be passed.        */
/*   - A node in the CRG's recovery domain is being ended with the        */
```

```
/*     QcstEndClusterNode API.  The node being ended will see the End Node */
/*     action code.  All other nodes in the recovery domain will see the   */
/*     Failover action code.  Action code dependent data of QcstEndNode    */
/*     will be passed for the Failover action code.                        */
/*   - An active recovery domain node for an active CRG is being removed   */
/*     from the cluster with the QcstRemoveClusterNodeEntry API.   Action  */
/*     code dependent data of QcstRemoveNode will be passed.  If an        */
/*     inactive node is removed for an active CRG, or if the CRG is        */
/*     inactive, an action code of Remove Node is passed.                  */
/*                                                                         */
/* The exit program is called regardless of whether or not the CRG is      */
/* active.  The exit program may have nothing to do if the CRG is not      */
/* active.                                                                 */
/*                                                                         */
/* If the CRG is active and the leaving member was the primary node,       */
/* perform the functions necessary for failover to a new primary.          */
/*                                                                         */
/* The Action_Code_Dependent_Data field can be used to determine if:       */
/*  - the failure was due to a problem that caused the cluster to become   */
/*     partitioned (all CRGs which had the partitioned nodes in the        */
/*     recovery domain are affected)                                       */
/*  - a node failed or had cluster resource services ended on the node (all*/
/*    CRGs which had the failed/ended node in the recovery  domain are     */
/*    affected)                                                            */
/*  - only a single CRG was affected (for example a single CRG job was     */
/*    canceled on a node or a single application failed)                  */
/*                                                                         */
/*                                                                         */
/* Things to consider:                                                     */
/*   - Prepare the new primary node so the application can be started.     */
/*   - The application should NOT be started at this time.  The exit       */
/*     program will be called again with the QcstCrgAcStart action code if */
/*     the CRG was active when the failure occurred.                       */
/*   - If the application CRG is active, ensure the required data CRGs are */
/*     active.                                                             */
/*                                                                         */
/*************************************************************************/
static int memberIsLeaving(int role,
                           int priorRole,
                           Qcst_EXTP0100_t *crgData,
                           EpData *epData) {


/*-----------------------------------------------------------------------*/
  /*                                                                     */
  /* If the CRG is active, perform failover.  Otherwise, nothing to do.  */
  /*                                                                     */

/*-----------------------------------------------------------------------*/
  if (crgData->Original_Cluster_Res_Grp_Stat == QcstCrgActive) {


/*---------------------------------------------------------------------*/
    /*                                                                 */
    /* The CRG is active.  Determine if my role has changed and I am now */
    /* the new primary.                                                */
    /*                                                                 */

/*---------------------------------------------------------------------*/

    if (priorRole != role && role == QcstPrimaryNodeRole) {


/*-------------------------------------------------------------------*/
      /*                                                             */
      /* I was not the primary but am now.  Do failover actions but don't */
      /* start the application at this time because this exit program will */
```

```
      /* be called again with the Start action code.                     */
      /*                                                                 */

/*-----------------------------------------------------------------------*/


/*-----------------------------------------------------------------------*/
      /*                                                                 */
      /* Ensure the data area status on this node starts out indicating  */
      /* the application is not running.                                 */
      /*                                                                 */

/*-----------------------------------------------------------------------*/
      setApplCrgDataArea(Appl_Ended);


/*-----------------------------------------------------------------------*/
      /*                                                                 */
      /* If the application has no actions to do on the Start action code */
      /* and will become active as soon as the takeover IP address is    */
      /* activated, then this code should be uncommented.  This code will */
      /* determine if all CRGs that this application CRG is dependent upon */
      /* are ready.  If this check fails, return failure from the action */
      /* code.                                                           */
      /*                                                                 */

/*-----------------------------------------------------------------------*/
/*      if (QcstSuccessful != checkDependCrgDataArea(MaxWaitSeconds))  */
/*         return QcstFailWithOutRestart;                              */

    }
  }

  return QcstSuccessful;
}  /* end memberIsLeaving()                                            */


/*************************************************************************/
/*                                                                       */
/* Action code = QcstCrgAcSwitchover                                     */
/*                                                                       */
/* The QcstInitiateSwitchOver API was called.  The first backup node in  */
/* the cluster resource group's recovery domain is taking over as the    */
/* primary node and the current primary node is being made the last backup.*/
/*                                                                       */
/* Things to consider:                                                   */
/*   - Prepare the new primary node so the application can be started.   */
/*   - The application should NOT be started at this time.  The exit     */
/*     program will be called again with the QcstCrgAcStart action code. */
/*   - The job running the application is canceled and the IP takeover  */
/*     address is ended prior to the exit program being called on the    */
/*     current primary.                                                  */
/*   - Ensure required data or device CRGs have switched over and are    */
/*     active.                                                           */
/*                                                                       */
/*************************************************************************/
static int switchPrimary(int role,
                         int priorRole,
                         Qcst_EXTP0100_t *crgData,
                         EpData *epData) {


/*-----------------------------------------------------------------------*/
  /*                                                                     */
  /* See if I am the old primary.                                        */
  /*                                                                     */
```

```
/*--------------------------------------------------------------------*/
  if (priorRole == QcstPrimaryNodeRole) {

/*--------------------------------------------------------------------*/
    /*                                                                */
    /* Do what ever needs to be done to cleanup the old primary before the */
    /* switch.  Remember that that job which was running the exit program  */
    /* which started the application was canceled already.               */
    /*                                                                */
    /* One example may be to clean up any processes holding locks on the   */
    /* database.  This may have been done by the application cancel       */
    /* handler if one was invoked.                                    */

/*--------------------------------------------------------------------*/
  }


/*--------------------------------------------------------------------*/
  /*                                                                  */
  /* I'm not the old primary.  See if I'm the new primary.            */
  /*                                                                  */

/*--------------------------------------------------------------------*/
  else if (role == QcstPrimaryNodeRole) {

/*--------------------------------------------------------------------*/
    /*                                                                */
    /* Do what ever needs to be done on the new primary before the        */
    /* application is started with the QcstCrgAcStart action code.        */
    /*                                                                */

/*--------------------------------------------------------------------*/


/*--------------------------------------------------------------------*/
    /*                                                                */
    /* Ensure the data area status on this nodes starts out indicating    */
    /* the application is not running.                                */
    /*                                                                */

/*--------------------------------------------------------------------*/
    setApplCrgDataArea(Appl_Ended);


/*--------------------------------------------------------------------*/
    /*                                                                */
    /* If the application has no actions to do on the Start action code   */
    /* and will become active as soon as the takeover IP address is       */
    /* activated, then this code should be uncommented.  This code will   */
    /* determine if all CRGs that this application CRG is dependent upon  */
    /* are ready.  If this check fails, return failure from the action    */
    /* code.                                                          */
    /*                                                                */

/*--------------------------------------------------------------------*/
/*      if (QcstSuccessful != checkDependCrgDataArea(MaxWaitSeconds))     */
/*         return QcstFailWithOutRestart;                              */

  }
  else {

/*--------------------------------------------------------------------*/
    /*                                                                */
    /* This node is one of the other backup nodes or it is a replicate    */
    /* node.  If there is anything those nodes must do, do it here.  If */
    /* not, remove this else block.                                   */
    /*                                                                */
```

```
/*--------------------------------------------------------------------*/


/*--------------------------------------------------------------------*/
    /*                                                                */
    /* Ensure the data area status on this nodes starts out indicating */
    /* the application is not running.                                */
    /*                                                                */

/*--------------------------------------------------------------------*/
    setApplCrgDataArea(Appl_Ended);
  }

  return QcstSuccessful;
}  /* end switchPrimary()                                            */


/**********************************************************************/
/*                                                                    */
/* Action code = QcstCrgAcAddNode                                     */
/*                                                                    */
/* The QcstAddNodeToRcvyDomain API was called.  A new node is being added */
/* to the recovery domain of a cluster resource group.                */
/*                                                                    */
/* Things to consider:                                                */
/*   - A new node is being added to the recovery domain.  See the     */
/*     considerations in the createCrg() function.                    */
/*   - If this CRG is enabled to use the QcstDistributeInformation API, */
/*     the user queue needed by that API could be created at this time. */
/*                                                                    */
/**********************************************************************/
static int addNode(int role,
                   int priorRole,
                   Qcst_EXTP0100_t *crgData,
                   EpData *epData) {


/*--------------------------------------------------------------------*/

  /*                                                                  */
  /* Determine if I am the node being added.                          */
  /*                                                                  */

/*--------------------------------------------------------------------*/

  if (0 == memcmp(&crgData->This_Nodes_ID,
                  &crgData->Changing_Node_ID,
                  sizeof(Qcst_Node_Id_t)))
  {

/*--------------------------------------------------------------------*/

    /*                                                                */
    /* Set the status of the data area on this new node.              */
    /*                                                                */

/*--------------------------------------------------------------------*/
    setApplCrgDataArea(Appl_Ended);


/*--------------------------------------------------------------------*/

    /*                                                                */
    /* Create the queue needed by the Distribute Information API.     */
    /*                                                                */
```

```
/*--------------------------------------------------------------------*/

    if (0 == memcmp(&crgData->DI_Queue_Name,
                    Nulls,
                    sizeof(crgData->DI_Queue_Name)))
    {
    }
  }

  return QcstSuccessful;
}  /* end addNode()
       */


/*****************************************************************************/
/*                                                                           */
/* Action code = QcstCrgAcRemoveNode                                         */
/*                                                                           */
/* The QcstRemoveNodeFromRcvyDomain or the QcstRemoveClusterNodeEntry        */
/* API was called.  A node is being removed from the recovery domain of      */
/* a cluster resource group or it is being removed entirely from the         */
/* cluster.                                                                  */
/*                                                                           */
/* This action code is seen by:                                              */
/*  For the QcstRemoveClusterNodeEntry API:                                  */
/*    - If the removed node is active and the CRG is Inactive, all nodes in*/
/*      the recovery domain including the node being removed see this        */
/*      action code.  The nodes NOT being removed see action code dependent*/
/*      data of QcstNodeFailure.                                             */
/*    - If the removed node is active and the CRG is Active, the node being*/
/*      removed sees the Remove Node action code.  All other nodes in the  */
/*      recovery domain see an action code of Failover and action code       */
/*      dependent data of QcstNodeFailure.                                   */
/*    - If the node being removed is not active in the cluster, all nodes  */
/*      in the recovery domain will see this action code.                    */
/*  For the QcstRemoveNodeFromRcvyDomain API:                                */
/*    - All nodes see the Remove Node action code regardless of whether or */
/*      not the CRG is Active.  Action code dependent data of                */
/*      QcstRmvRcvyDmnNode will also be passed.                              */
/*                                                                           */
/* Things to consider:                                                       */
/*   - You may want to cleanup the removed node by deleting objects no       */
/*     longer needed there.                                                  */
/*   - The job running the application is canceled and the IP takeover    */
/*     address is ended after the exit program is called if this is the      */
/*     primary node and the CRG is active.                                   */
/*   - If subsystems or server jobs were started as a result of the          */
/*     QcstCrgAcStart action code, end them here or consolidate all logic  */
/*     to end the application in the cancelHandler() since it will be        */
/*     invoked for all Cluster Resource Services APIs which must end the     */
/*     application on the current primary.                                   */
/*                                                                           */
/*****************************************************************************/
static int rmvNode(int role,
                   int priorRole,
                   Qcst_EXTP0100_t *crgData,
                   EpData *epData) {


/*--------------------------------------------------------------------*/

  /*                                                                   */
  /* Determine if I am the node being removed.                         */
  /*                                                                   */

/*--------------------------------------------------------------------*/
```

```
  if (0 == memcmp(&crgData->This_Nodes_ID,
                  &crgData->Changing_Node_ID,
                  sizeof(Qcst_Node_Id_t)))
  {

/*--------------------------------------------------------------------*/
      /*                                                              */
      /* End the application if it is running on this node.           */
      /*                                                              */

/*--------------------------------------------------------------------*/
      endApplication(QcstCrgAcRemoveNode, role, priorRole, crgData,
epData);

  }
  return QcstSuccessful;
}  /* end rmvNode                                                     */


/************************************************************************/
/*                                                                      */
/* Action code = QcstCrgAcChange                                        */
/*                                                                      */
/* The QcstChangeClusterResourceGroup API was called.  Some attribute   */
/* or information stored in the cluster resource group object is being  */
/* changed.  Note that not all changes to the CRG object cause the exit */
/* program to be called.  As of V5R1M0, only these changes will cause the */
/* exit program to be called-                                           */
/*   - the current recovery domain is being changed                     */
/*   - the preferred recovery domain is being changed                   */
/*                                                                      */
/* If any of the above changes are being made but additionally the exit */
/* program is being changed to *NONE, the exit program is not called.   */
/*                                                                      */
/* Things to consider:                                                  */
/*   - None unless changing the recovery domain affects information or  */
/*     processes for this cluster resource group.  Note that the primary */
/*     node cannot be changed with the QcstChangeClusterResourceGroup API */
/*     if the CRG is active.                                            */
/*                                                                      */
/************************************************************************/
static int chgCrg(int role,
                  int priorRole,
                  Qcst_EXTP0100_t *crgData,
                  EpData *epData) {

  return QcstSuccessful;
}  /* end chgCrg()                                                    */


/************************************************************************/
/*                                                                      */
/* Action code = QcstCrgAcDeleteCommand                                 */
/*                                                                      */
/* The Delete Cluster Resource Group (DLTCRG) CL command has been called */
/* to delete a cluster resource group object, the QcstDeleteCluster API */
/* has been called, or the QcstRemoveClusterNodeEntry API has been called. */
/* In each case, cluster resource services is not active on the cluster */
/* node where the command or API was called.  Thus, this function is not */
/* distributed cluster wide but occurs only on the node where the CL    */
/* command or API was called.                                           */
/*                                                                      */
/* If the QcstDeleteCluster API was used, action code dependent data of */
/* QcstDltCluster is passed.                                            */
/*                                                                      */
/* See the considerations in the deleteCrg() function                   */
/*                                                                      */
```

```
/*****************************************************************************/
static int deleteCrgWithCmd(int role,
                            int doesNotApply,
                            Qcst_EXTP0100_t *crgData,
                            EpData *epData) {

  return QcstSuccessful;
}  /* end deleteCrgWithCmd()                                                 */


/*****************************************************************************/
/*                                                                           */
/* Action code = QcstCrgEndNode                                              */
/*                                                                           */
/* The QcstEndClusterNode API was called or a CRG job was canceled.       */
/*                                                                           */
/* The QcstCrgEndNode action code is passed to the exit program only on the*/
/* node being ended or where the CRG job was canceled.  On the node where */
/* a Cluster Resource Services job is canceled, action code dependent data*/
/* of QcstMemberFailure will be passed.                                      */
/* When Cluster Resource Services ends on this node or the CRG job ends, it*/
/* will cause all other nodes in the cluster to go through failover        */
/* processing.  The action code passed to all other nodes will be          */
/* QcstCrgAcFailover.   Those nodes will see action code dependent data of */
/* QcstMemberFailure if a CRG job is canceled or QcstNodeFailure if the   */
/* node is ended.                                                            */
/*                                                                           */
/* Things to consider:                                                       */
/*   - The job running the application is canceled and the IP takeover    */
/*     address is ended after the exit program is called if this is the    */
/*     primary node and the CRG is active.                                   */
/*   - If subsystems or server jobs were started as a result of the        */
/*     QcstCrgAcStart action code, end them here.                            */
/*                                                                           */
/*****************************************************************************/
static int endNode(int role,
                   int priorRole,
                   Qcst_EXTP0100_t *crgData,
                   EpData *epData) {


/*---------------------------------------------------------------------------*/
  /*                                                                         */
  /* End the application if it is running on this node.                      */
  /*                                                                         */

/*---------------------------------------------------------------------------*/
  endApplication(QcstCrgEndNode, role, priorRole, crgData, epData);

  return QcstSuccessful;
}  /* end endNode()                                                          */


/*****************************************************************************/
/*                                                                           */
/* Action code = QcstCrgAcChgNodeStatus                                      */
/*                                                                           */
/* The QcstChangeClusterNodeEntry API was called.  The status of a node    */
/* is being changed to failed.  This API is used to inform cluster resource*/
/* services that the node did not partition but really failed.             */
/*                                                                           */
/* Things to consider:                                                       */
/*   - The exit program was called previously with an action code of       */
/*     QcstCrgAcEnd if the CRG was active or an action code of             */
/*     QcstCrgAcFailover if the CRG was inactive because cluster resource  */
/*     services thought the cluster had become partitioned.  The user is   */
/*     now telling cluster resource services that the node really failed   */
```

```
/*     instead of partitioned.  The exit program has something to do only  */
/*     if it performed some action previously that needs to be changed now */
/*     that node failure can be confirmed.                                 */
/*                                                                         */
/***************************************************************************/
static int chgNodeStatus(int role,
                         int priorRole,
                         Qcst_EXTP0100_t *crgData,
                         EpData *epData) {

  return QcstSuccessful;
}  /* end chgNodeStatus()                                                  */


/***************************************************************************/
/*                                                                         */
/* Action code = QcstCrgAcCancelFailover                                   */
/*                                                                         */
/* Cluster resource services on the primary node has failed or ended       */
/* for this cluster resource group.  A message was sent to the failover    */
/* message queue specified for the CRG, and the result of that message     */
/* was to cancel the failover.  This will change the status of the CRG to  */
/* inactive and leave the primary node as primary.                         */
/*                                                                         */
/* Things to consider:                                                     */
/*   - The primary node is no longer participating in cluster activities.  */
/*     The problem which caused the primary node to fail should be fixed   */
/*     so that the CRG may be started again.                               */
/*                                                                         */
/***************************************************************************/
static int cancelFailover(int role,
                          int priorRole,
                          Qcst_EXTP0100_t *crgData,
                          EpData *epData) {

  return QcstSuccessful;
}  /* end cancelFailover()                                                 */


/***************************************************************************/
/*                                                                         */
/* Action code = exit program does not know it yet                         */
/*                                                                         */
/* A new action code has been passed to this exit program.  This can occur */
/* after a new i5/OS release has been installed and some new cluster API  */
/* was called or some new cluster event occurred.  The logic in this exit  */
/* program has not yet been updated to understand the new action code.     */
/*                                                                         */
/* Two different strategies could be used for the new action code.  The    */
/* correct strategy is dependent upon the kinds of things this particular  */
/* exit program does for the application.                                  */
/*                                                                         */
/* One strategy is to not do anything and return a successful return code. */
/* This allows the new cluster API or event to run to completion.  It      */
/* allows the function to be performed even though this exit program       */
/* did not understand the new action code.  The risk, though, is that the  */
/* exit program should have done something and it did not.  At a minimum,  */
/* you may want to log some kind of error message about what happened so   */
/* that programming can investigate and get the exit program updated.      */
/*                                                                         */
/* The opposite strategy is to return an error return code such as         */
/* QcstFailWithRestart.  Of course doing this means that the new cluster   */
/* API or event cannot be used until the exit program is updated for the   */
/* new action code.  Again, logging some kind of error message for         */
/* programming to investigate would be worthwhile.                         */
/*                                                                         */
/* Only the designer of the exit program can really decide which is the    */
```

```
/* better course of action.                                                  */
/*                                                                           */
/*****************************************************************************/
static int newActionCode(int role,
                         int doesNotApply,
                         Qcst_EXTP0100_t *crgData,
                         EpData *epData) {


/*---------------------------------------------------------------------------*/
  /*                                                                         */
  /* Add logic to log an error somewhere - operator message queue, job       */
  /* log, application specific error log, etc. so that the exit program      */
  /* gets updated to properly handle the new action code.                    */
  /*                                                                         */
  /* Note that if this is left coded as it is, this is the "don't do         */
  /* anything" strategy described in the prologue above.                     */
  /*                                                                         */

/*---------------------------------------------------------------------------*/

  return QcstSuccessful;
}  /* end newActionCode()                                                    */


/*****************************************************************************/
/*                                                                           */
/* Action code = QcstCrgAcUndo                                               */
/*                                                                           */
/* Note: The exit program is never called with an undo action code for       */
/* any of these prior action codes:                                          */
/*   QcstCrgAcChgNodeStatus                                                  */
/*   QcstCrgAcDelete                                                         */
/*   QcstCrgAcDeleteCommand                                                  */
/*   QcstCrgEndNode                                                          */
/*   QstCrgAcRemoveNode (If the node being removed is active in the          */
/*                       cluster and the API is Remove Cluster Node.         */
/*                       The Remove Node From Recovery Domain will call      */
/*                       with Undo and the Remove Cluster Node API will      */
/*                       call with Undo if the node being removed is         */
/*                       inactive.                                           */
/*   QcstCrgAcRestart                                                        */
/*   QcstCrgAcUndo                                                           */
/*                                                                           */
/* APIs that call an exit program do things in 3 steps.                      */
/*   1. Logic which must be done prior to calling the exit program.          */
/*   2. Call the exit program.                                               */
/*   3. Logic which must be done after calling the exit program.             */
/*                                                                           */
/* Any errors that occur during steps 2 or 3 result in the exit program      */
/* being called again with the undo action code.  This gives the exit        */
/* program an opportunity to back out any work performed when it was first   */
/* called by the API.  The API will also be backing out any work it          */
/* performed trying to return the state of the cluster and cluster objects   */
/* to what it was before the API was called.                                 */
/*                                                                           */
/* It is suggested that the following return codes be returned for the       */
/* specified action code as that return code will result in the most         */
/* appropriate action being taken.                                           */
/*                                                                           */
/*   QcstCrgAcInitialize: QcstSuccessful; The CRG is not created.            */
/*   QcstCrgAcStart:      QcstSuccessful; The CRG is not started.            */
/*   QcstCrgAcEnd:        QcstFailWithOutRestart; The CRG is set to Indoubt*/
/*                                        The cause of the failure needs to*/
/*                                        investigated.                    */
/*   QcstCrgAcReJoin:     QcstFailWithOutRestart; The CRG is set to Indoubt*/
/*                                        The cause of the failure needs to*/
```

```
/*                                      investigated.                   */
/*   QcstCrgAcFailover:   QcstFailWithOutRestart; The CRG is set to Indoubt*/
/*                                      The cause of the failure needs to*/
/*                                      investigated.                   */
/*   QcstCrgAcSwitchover: QcstFailWithOutRestart; The CRG is set to Indoubt*/
/*                                      The cause of the failure needs to*/
/*                                      investigated.                   */
/*   QcstCrgAcAddNode:    QcstSuccessful; The node is not added.         */
/*   QcstCrgAcRemoveNode: QcstFailWithOutRestart; The CRG is set to Indoubt*/
/*                                      The cause of the failure needs to*/
/*                                      investigated.                   */
/*   QcstCrgAcChange:     QcstSuccessful; The recovery domain is not     */
/*                                      changed.                        */
/*                                                                      */
/************************************************************************/
static int undoPriorAction(int role,
                           int priorRole,
                           Qcst_EXTP0100_t *crgData,
                           EpData *epData) {


/*----------------------------------------------------------------------*/
  /*                                                                    */
  /* The prior action code defines what the exit program was doing when */
  /* it failed, was canceled, or returned a non successful return code. */
  /*                                                                    */

/*----------------------------------------------------------------------*/
  if (crgData->Prior_Action_Code &lt;= MaxAc )
    return (*undoFcn[crgData-&lt;Prior_Action_Code])
                                       (role, priorRole, crgData,
epData);
  else

/*----------------------------------------------------------------------*/
    /*                                                                  */
    /* IBM has defined a new action code in a new operating system release */
    /* and this exit program has not yet been updated to handle it.  Take a*/
    /* default action for now.                                          */
    /*                                                                  */

/*----------------------------------------------------------------------*/
    return newActionCode(role, priorRole, crgData, epData);
}  /* end undoPriorAction()                                             */


/************************************************************************/
/*                                                                      */
/* Action code = QcstCrgAcUndo                                          */
/*                                                                      */
/* Prior action code = QcstCrgAcInitialize                              */
/*                                                                      */
/* Things to consider:                                                  */
/*   The CRG will not be created.  Objects that might have been created */
/*   on nodes in the recovery domain should be deleted since a subsequent */
/*   create could fail if those objects already exist.                  */
/*                                                                      */
/************************************************************************/
static int undoCreateCrg(int role,
                         int doesNotApply,
                         Qcst_EXTP0100_t *crgData,
                         EpData *epData) {

  return QcstSuccessful;
}  /* end undoCreateCrg()                                               */
```

```
/*********************************************************************/
/*                                                                   */
/* Action code = QcstCrgAcUndo                                       */
/*                                                                   */
/* Prior action code = QcstCrgAcStart                                */
/*                                                                   */
/* Things to consider:                                               */
/*   Cluster Resource Services failed when it was finishing the Start CRG  */
/*   API after it had already called the exit program with the Start       */
/*   Action code.                                                    */
/*                                                                   */
/*   On the primary node, the exit program job which is running the        */
/*   application will be canceled.  The exit program will then be called  */
/*   with the Undo action code.                                      */
/*                                                                   */
/*   All other nodes in the recovery domain will be called with the Undo   */
/*   action code.                                                    */
/*                                                                   */
/*********************************************************************/
static int undoStartCrg(int role,
                        int doesNotApply,
                        Qcst_EXTP0100_t *crgData,
                        EpData *epData) {

  return QcstSuccessful;
}  /* end undoStartCrg()                                             */


/*********************************************************************/
/*                                                                   */
/* Action code = QcstCrgAcUndo                                       */
/*                                                                   */
/* Prior action code = QcstCrgAcEnd                                  */
/*                                                                   */
/* Things to consider:                                               */
/*   The CRG will not be ended.  If the exit program did anything to bring */
/*   down the application it can either restart the application or it can  */
/*   decide to not restart the application.  If the application is not     */
/*   restarted, the return code should be set to QcstFailWithOutRestart so */
/*   the status of the CRG is set to Indoubt.                        */
/*                                                                   */
/*********************************************************************/
static int undoEndCrg(int role,
                      int doesNotApply,
                      Qcst_EXTP0100_t *crgData,
                      EpData *epData) {

  return QcstFailWithOutRestart;
}  /* end undoEndCrg()                                               */


/*********************************************************************/
/*                                                                   */
/* Action code = QcstCrgAcUndo                                       */
/*                                                                   */
/* Prior action code = QcstCrgAcReJoin                               */
/*                                                                   */
/* Things to consider:                                               */
/*   An error occurred which won't allow the member to join this CRG       */
/*   group.  Anything done for the Join action code needs to be looked at  */
/*   to see if something must be undone if this member is not an active    */
/*   member of the CRG group.                                        */
/*                                                                   */
/*********************************************************************/
static int undoMemberIsJoining(int role,
                               int doesNotApply,
                               Qcst_EXTP0100_t *crgData,
```

```
                              EpData *epData) {

  return QcstFailWithOutRestart;
}  /* end undoMemberIsJoining()                                          */


/*************************************************************************/
/*                                                                       */
/* Action code = QcstCrgAcUndo                                           */
/*                                                                       */
/* Prior action code = QcstCrgAcFailover                                 */
/*                                                                       */
/* Things to consider:                                                   */
/*   This does not mean that the node failure or failing member is being */
/*   undone.  That failure is irreversible.  What it does mean is that the */
/*   exit program returned an error from the Failover action code or     */
/*   Cluster Resource Services ran into a problem after it called the exit */
/*   program.  If the CRG was active when Failover was attempted, it is  */
/*   not at this point.  End the resilient resource and expect a human to */
/*   look into the failure.  After the failure is corrected, the CRG will */
/*   must be started with the Start CRG API.                          */
/*                                                                       */
/*                                                                       */
/*************************************************************************/
static int undoMemberIsLeaving(int role,
                               int doesNotApply,
                               Qcst_EXTP0100_t *crgData,
                               EpData *epData) {

  return QcstFailWithOutRestart;
}  /* end undoMemberIsLeaving()                                          */


/*************************************************************************/
/*                                                                       */
/* Action code = QcstCrgAcUndo                                           */
/*                                                                       */
/* Prior action code = QcstCrgAcSwitchover                               */
/*                                                                       */
/* Things to consider:                                                   */
/*   Some error occurred after the point of access was moved from the    */
/*   original primary and before it could be brought up on the new primary.*/
/*   The IP address was ended on the original primary before moving the  */
/*   point of access but is started on the original primary again.  Cluster*/
/*   Resource Services will now attempt to move the point of access back */
/*   to the original primary.  The application exit program and IP takeover*/
/*   address will be started on the original primary.                    */
/*                                                                       */
/*                                                                       */
/*************************************************************************/
static int undoSwitchPrimary(int role,
                             int doesNotApply,
                             Qcst_EXTP0100_t *crgData,
                             EpData *epData) {

  return QcstFailWithOutRestart;
}  /* end undoSwitchPrimary()                                            */


/*************************************************************************/
/*                                                                       */
/* Action code = QcstCrgAcUndo                                           */
/*                                                                       */
/* Prior action code = QcstCrgAcAddNode                                  */
/*                                                                       */
/* Things to consider:                                                   */
/*   If objects were created on the new node, they should be removed so  */
```

```
/*   that a subsequent Add Node to aRecovery Domain does not fail if it    */
/*   attempts to create objects again.                                      */
/*                                                                          */
/*                                                                          */
/****************************************************************************/
static int undoAddNode(int role,
                       int doesNotApply,
                       Qcst_EXTP0100_t *crgData,
                       EpData *epData) {

  return QcstSuccessful;
}  /* end undoAddNode()                                                     */


/****************************************************************************/
/*                                                                          */
/* Action code = QcstCrgAcUndo                                              */
/*                                                                          */
/* Prior action code = QcstCrgAcRemoveNode                                  */
/*                                                                          */
/* Things to consider:                                                      */
/*   The node is still in the recovery domain.  If objects were removed    */
/*   from the node, they should be added back.                              */
/*                                                                          */
/****************************************************************************/
static int undoRmvNode(int role,
                       int doesNotApply,
                       Qcst_EXTP0100_t *crgData,
                       EpData *epData) {

  return QcstFailWithOutRestart;
}  /* end undoRmvNode()                                                     */


/****************************************************************************/
/*                                                                          */
/* Action code = QcstCrgAcUndo                                              */
/*                                                                          */
/* Prior action code = QcstCrgAcChange                                      */
/*                                                                          */
/* Things to consider:                                                      */
/*   Changes to the CRG will be backed out so that the CRG and its          */
/*   recovery domain look just like it did prior to the attempted change.   */
/*   Any changes the exit program made should also be backed out.           */
/*                                                                          */
/****************************************************************************/
static int undoChgCrg(int role,
                      int doesNotApply,
                      Qcst_EXTP0100_t *crgData,
                      EpData *epData) {

  return QcstSuccessful;
}  /* end undoChgCrg()                                                      */


/****************************************************************************/
/*                                                                          */
/* Action code = QcstCrgAcUndo                                              */
/*                                                                          */
/* Prior action code = QcstCrgAcCancelFailover                              */
/*                                                                          */
/* Things to consider:                                                      */
/*   This does not mean that the node failure or failing member is being    */
/*   undone.  That failure is irreversible.  What it does mean is that      */
/*   Cluster Resource Services ran into a problem after it called the exit  */
/*   program.  The CRG will be InDoubt regardless of what is returned from  */
/*   this exit program call.  Someone will need to manually look into the   */
```

```
/*   the failure.  After the failure is corrected, the CRG will must be */
/*   started with the Start CRG API.                                    */
/*                                                                      */
/*                                                                      */
/************************************************************************/
static int undoCancelFailover(int role,
                              int doesNotApply,
                              Qcst_EXTP0100_t *crgData,
                              EpData *epData) {

  return QcstSuccessful;
}  /* end undoCancelFailover()                                          */


/************************************************************************/
/*                                                                      */
/* A simple routine to take a null terminated object name and a null    */
/* terminated library name and build a 20 character non-null terminated */
/* qualified name.                                                      */
/*                                                                      */
/************************************************************************/
static void bldDataAreaName(char *objName, char* libName, char *qualName) {

  memset(qualName, 0x40, 20);
  memcpy(qualName, objName, strlen(objName));
  qualName += 10;
  memcpy(qualName, libName, strlen(libName));
  return;
}  /* end bldDataAreaName                                               */


/************************************************************************/
/*                                                                      */
/* The data area is checked to see if all the CRGs that this application */
/* is dependent upon are ready.  If they are not ready, a wait for a    */
/* certain amount of time is performed and the data area is checked again. */
/* This check, wait loop continues until all dependent CRGs become ready or*/
/* until the maximum wait time has been reached.                        */
/* The length of the wait can be changed to some other value if a       */
/* particular situation would be better with shorter or longer wait times. */
/*                                                                      */
/*                                                                      */
/************************************************************************/
static int checkDependCrgDataArea(unsigned int maxWaitTime) {

  Qus_EC_t errCode = { sizeof(Qus_EC_t), 0 };
  char dataAreaName[20];
  struct {
    Qwc_Rdtaa_Data_Returned_t stuff;
    char ready;
  } data;


/*--------------------------------------------------------------------*/
  /*                                                                  */
  /* This is an accumulation of the time waited for the dependent CRGs to  */
  /* become ready.                                                    */
  /*                                                                  */

/*--------------------------------------------------------------------*/
  unsigned int timeWaited = 0;


/*--------------------------------------------------------------------*/
  /*                                                                  */
  /* Build definition of the amount of time to wait.                  */
  /*                                                                  */
```

```
/*--------------------------------------------------------------------*/
  _MI_Time   timeToWait;
  int hours     = 0;
  int minutes   = 0;
  int seconds   = WaitSecondsIncrement;
  int hundreths = 0;
  short int options = _WAIT_NORMAL;
  mitime( &timeToWait, hours, minutes, seconds, hundreths );


/*--------------------------------------------------------------------*/
  /*                                                                  */
  /* Build the qualified name of the data area.                       */
  /*                                                                  */

/*--------------------------------------------------------------------*/
  bldDataAreaName(DependCrgDataArea, ApplLib, dataAreaName);


/*--------------------------------------------------------------------*/
  /*                                                                  */
  /* Get the data from the data area that indicates whether or not the     */
  /* CRGs are all ready.  This data area is updated by the High       */
  /* Availability Business Partners when it is ok for the application to   */
  /* proceed.                                                         */
  /*                                                                  */

/*--------------------------------------------------------------------*/
  QWCRDTAA(&data,
           sizeof(data),
           dataAreaName,
           offsetof(Qcst_HAAPPO_t,Data_Status)+1,  /* API wants a 1 origin */
           sizeof(data.ready),
           &errCode);


/*--------------------------------------------------------------------*/
  /*                                                                  */
  /* If the dependent CRGs are not ready, wait for a bit and check again.  */
  /*                                                                  */

/*--------------------------------------------------------------------*/
  while (data.ready != Data_Available) {


/*--------------------------------------------------------------------    */
    /*                                                                */
    /* If the dependent CRGs are not ready after the maximum wait time,    */
    /* return an error.  Consider logging some message to describe why the */
    /* application did not start so that  the problem can be looked into.  */
    /*                                                                */

/*--------------------------------------------------------------------*/
    if (timeWaited >= maxWaitTime)
      return QcstFailWithOutRestart;


/*--------------------------------------------------------------------*/
    /*                                                                  */
    /* Wait to allow the data CRGs to become ready.                     */
    /*                                                                  */

/*--------------------------------------------------------------------*/
    waittime(&timeToWait, options);
    timeWaited += WaitSecondsIncrement;
```

```
/*--------------------------------------------------------------------*/
    /*                                                                  */
    /* Get information from the data area again to see if the data CRGs are*/
    /* ready.                                                           */
    /*                                                                  */

/*--------------------------------------------------------------------*/
    QWCRDTAA(&data,
             sizeof(data),
             dataAreaName,
             offsetof(Qcst_HAAPPO_t,Data_Status)+1,  /* API wants a 1 origin */
             sizeof(data.ready),
             &errCode);
  }

  return QcstSuccessful;
}  /* end checkDependCrgDataArea                                      */


/**********************************************************************/
/*                                                                    */
/* The application CRG data area is updated to indicate that the      */
/* application is running or to indicate it is not running.  This data area*/
/* information is used by the High Availability Business Partners to  */
/* coordinate the switchover activities between CRGs that have dependencies*/
/* on each other.                                                     */
/*                                                                    */
/**********************************************************************/
static void setApplCrgDataArea(char status) {

  char cmd[54];
  char cmdEnd[3] = {0x00, ')', 0x00};


/*--------------------------------------------------------------------*/
  /*                                                                  */
  /* Set up the CL command string with the data area library name, the data*/
  /* area name, and the character to put into the data area.  Then run the */
  /* CL command.                                                      */
  /*                                                                  */

/*--------------------------------------------------------------------*/
  memcpy(cmd, "CHGDTAARA DTAARA(", strlen("CHGDTAARA DTAARA(")+1);
  strcat(cmd, ApplLib);
  strcat(cmd, "/");
  strcat(cmd, ApplCrgDataArea);
  strcat(cmd, " (425 1)) VALUE(");                          /* @A1C */
  cmdEnd[0] = status;
  strcat(cmd, cmdEnd);

  system(cmd);

  return;
}  /* end setApplCrgDataArea                                          */


/**********************************************************************/
/*                                                                    */
/* This function is called any time the exit program receives an exception */
/* not specifically monitored for by some other exception handler.  Add    */
/* appropriate logic to perform cleanup functions that may be required.    */
/* A failure return code is then set and control returns to the operating  */
/* system.  The job this exit program is running in will then end.         */
/*                                                                    */
/* When this function gets called, myData->role may still contain the      */
/* UnknownRole value if an exception occurred before this node's role      */
```

```
/* value was set.  To be completely correct, the role should be tested     */
/* for UnknownRole before making any decisions based upon the value of     */
/* role.                                                                   */
/*                                                                         */
/***************************************************************************/
static void unexpectedExceptionHandler(_INTRPT_Hndlr_Parms_T
*exData) {


/*-------------------------------------------------------------------------  */
  /*                                                                       */
  /* Get a pointer to the structure containing data that is passed to the  */
  /* exception handler.                                                    */
  /*                                                                       */

/*-------------------------------------------------------------------------*/
  HandlerDataT *myData = (HandlerDataT *)exData->Com_Area;


/*-------------------------------------------------------------------------*/
  /*                                                                       */
  /* Perform as much cleanup function as necessary.  Some global state     */
  /* information may must be kept so the exception handler knows what   */
  /* steps were completed before the failure occurred and thus knows what  */
  /* cleanup steps must be performed.  This state information could be     */
  /* kept in the HandlerDataT structure or it could be kept in some other  */
  /* location that this function can address.                              */
  /*                                                                       */

/*-------------------------------------------------------------------------*/


/*-------------------------------------------------------------------------*/
  /*                                                                       */
  /* If this is the primary node and the application was started, end it.  */
  /* The application is ended because the exit program will be called again*/
  /* with the Restart action code and want the restartCrg() function to */
  /* always work the same way.  In addition, ending the application may    */
  /* clear up the condition that caused the exception.    */
  /* If possible, warn users and have them stop using the application so   */
  /* things are done in an orderly manner.                                 */
  /*                                                                       */

/*-------------------------------------------------------------------------*/
  endApplication(myData->actionCode,
                 myData->role,
                 myData->priorRole,
                 myData->crgData,
                 myData->epData);


/*-------------------------------------------------------------------------*/
  /*                                                                       */
  /* Set the exit program return code.                                     */
  /*                                                                       */

/*-------------------------------------------------------------------------*/
  *myData->retCode = QcstFailWithRestart;


/*-------------------------------------------------------------------------*/
  /*                                                                       */
  /* Let the exception percolate up the call stack.                 */
  /*                                                                       */

/*-------------------------------------------------------------------------*/
  return;
```

```
}  /* end unexpectedExceptionHandler                                        */


/*************************************************************************/
/*                                                                       */
/* This function is called any time the job this exit program is running in*/
/* is canceled.  The job could be canceled due to any of the following   */
/* (the list is not intended to be all inclusive)-                       */
/*   - an API cancels an active application CRG.  The End CRG, Initiate    */
/*     Switchover, End Cluster Node, Remove Cluster Node or Delete Cluster */
/*     API cancels the job which was submitted when the exit program was   */
/*     called with a Start action code.                                  */
/*   - operator cancels the job from some operating system display such as */
/*     Work with Active Jobs                                             */
/*   - the subsystem this job is running in is ended                     */
/*   - all subsystems are ended                                          */
/*   - the system is powered down                                        */
/*   - an operating system machine check occurred                        */
/*                                                                       */
/* When this function gets called, myData->role may still contain the      */
/* UnknownRole value if cancelling occurred before this node's role        */
/* value was set.  To be completely correct, the role should be tested     */
/* for UnknownRole before making any decisions based upon the value of     */
/* role.                                                                 */
/*                                                                       */
/*************************************************************************/
static void cancelHandler(_CNL_Hndlr_Parms_T *cnlData) {


/*-----------------------------------------------------------------------*/
  /*                                                                     */
  /* Get a pointer to the structure containing data that was passed to the */
  /* cancel handler.                                                     */
  /*                                                                     */

/*-----------------------------------------------------------------------*/
  HandlerDataT *myData = (HandlerDataT *)cnlData->Com_Area;


/*-----------------------------------------------------------------------*/
  /*                                                                     */
  /* Perform as much cleanup function as necessary.  Some global state     */
  /* information may must be kept so the cancel handler knows what      */
  /* steps were completed before the job was canceled and thus knows if   */
  /* the function had really completed successfully or was only partially  */
  /* complete and thus needs some cleanup to be done.  This state          */
  /* information could be kept in the HandlerDataT structure or it could   */
  /* be kept in some other location that this function can address.        */
  /*                                                                     */

/*-----------------------------------------------------------------------*/


/*-----------------------------------------------------------------------*/

  /*                                                                     */
  /* This job is being canceled.  If I was running the application as a   */
  /* result of the Start or Restart action codes, end the application now. */
  /* This job is being canceled because a Switch Over or some other       */
  /* Cluster Resource Services API was used which affects the primary node */
  /* or someone did a cancel job with a CL command, from a system display, */
  /* etc.                                                                */


/*-----------------------------------------------------------------------*/
  endApplication(myData->actionCode,
                 myData->role,
```

```
                myData->priorRole,
                myData->crgData,
                myData->epData);


/*--------------------------------------------------------------------*/
  /*                                                                  */
  /* Set the exit program return code.                                */
  /*                                                                  */

/*--------------------------------------------------------------------*/
  *myData->retCode = QcstSuccessful;


/*--------------------------------------------------------------------*/
  /*                                                                  */
  /* Return to the operating system for final ending of the job.      */
  /*                                                                  */

/*--------------------------------------------------------------------*/
  return;
}  /* end cancelHandler                                             */


/********************************************************************/
/*                                                                  */
/* A common routine used to end the application by various action code     */
/* functions, the exception handler, and the cancel handler.        */
/*                                                                  */
/********************************************************************/
static void endApplication(unsigned int actionCode,
                           int role,
                           int priorRole,
                           Qcst_EXTP0100_t *crgData,
                           EpData *epData) {

  if (  role == QcstPrimaryNodeRole
      &&
        crgData->Original_Cluster_Res_Grp_Stat == QcstCrgActive)
{

/*--------------------------------------------------------------------*/
    /*                                                                */
    /* Add logic to end the application here.  You may need to add logic   */
    /* to determine if the application is still running because this       */
    /* function could be called once for an action code and again from    */
    /* the cancel handler (End CRG is an example).                    */
    /*                                                                */

/*--------------------------------------------------------------------*/




/*--------------------------------------------------------------------*/
    /*                                                                */
    /* After the application has ended, update the data area to indicate   */
    /* the application is no longer running.                          */
    /*                                                                */

/*--------------------------------------------------------------------*/
    setApplCrgDataArea(Appl_Ended);
  }

  return;
}  /* end endApplication                                            */
```

```
/*****************************************************************************/
/*                                                                           */
/* Print out the data passed to this program.                                */
/*                                                                           */
/*****************************************************************************/
static void printParms(int actionCode,
                       int role,
                       int priorRole,
                       Qcst_EXTP0100_t *crgData,
                       EpData *epData) {

  unsigned int i;
  char *str;

  /* Print the action code.                                                  */
  printf("%s", "Action_Code = ");
  printActionCode(actionCode);

  /* Print the action code dependent data.                                   */
  printf("%s", "    Action_Code_Dependent_Data = ");
  switch (crgData->Action_Code_Dependent_Data) {
    case QcstNoDependentData:  str = "QcstNoDependentData";
                               break;
    case QcstMerge:            str = "QcstMerge";
                               break;
    case QcstJoin:             str = "QcstJoin";
                               break;
    case QcstPartitionFailure: str = "QcstPartitionFailure";
                               break;
    case QcstNodeFailure:      str = "QcstNodeFailure";
                               break;
    case QcstMemberFailure:    str = "QcstMemberFailure";
                               break;
    case QcstEndNode:          str = "QcstEndNode";
                               break;
    case QcstRemoveNode:       str = "QcstRemoveNode";
                               break;
    case QcstApplFailure:      str = "QcstApplFailure";
                               break;
    case QcstResourceEnd:      str = "QcstResourceEnd";
                               break;
    case QcstDltCluster:       str = "QcstDltCluster";
                               break;
    case QcstRmvRcvyDmnNode:   str = "QcstRmvRcvyDmnNode";
                               break;
    case QcstDltCrg:           str = "QcstDltCrg";
                               break;
    default: str = "unknown action code dependent data";
  }
  printf("%s \n", str);


  /* Print the prior action code.                                            */
  printf("%s", "  Prior_Action_Code = ");
  if (crgData->Prior_Action_Code)
    printActionCode(crgData->Prior_Action_Code);
  printf("\n");

  /* Print the cluster name.                                                 */
  printStr("  Cluster_Name = ",
           crgData->Cluster_Name, sizeof(Qcst_Cluster_Name_t));

  /* Print the CRG name.                                                     */
  printStr("  Cluster_Resource_Group_Name = ",
           crgData->Cluster_Resource_Group_Name,
sizeof(Qcst_Crg_Name_t));
```

```
  /* Print the CRG type.                                              */
  printf("%s \n", "  Cluster_Resource_Group_Type =
QcstCrgApplResiliency");

  /* Print the CRG status.                                            */
  printf("%s", "  Cluster_Resource_Group_Status = ");
  printCrgStatus(crgData->Cluster_Resource_Group_Status);

  /* Print the CRG original status.                                   */
  printf("%s", "  Original_Cluster_Res_Grp_Stat = ");
  printCrgStatus(crgData->Original_Cluster_Res_Grp_Stat);

  /* Print the Distribute Information queue name.                     */
  printStr("  DI_Queue_Name = ",
           crgData->DI_Queue_Name,
sizeof(crgData->DI_Queue_Name));
  printStr("  DI_Queue_Library_Name = ",
           crgData->DI_Queue_Library_Name,
           sizeof(crgData->DI_Queue_Library_Name));

  /* Print the CRG attributes.                                        */
  printf("%s", "  Cluster_Resource_Group_Attr = ");
  if (crgData->Cluster_Resource_Group_Attr &
QcstTcpConfigByUsr)
    printf("%s", "User Configures IP Takeover Address");
  printf("\n");

  /* Print the ID of this node.                                       */
  printStr("  This_Nodes_ID = ",
           crgData->This_Nodes_ID, sizeof(Qcst_Node_Id_t));

  /* Print the role of this node.                                     */
  printf("%s %d \n", "  this node's role = ", role);

  /* Print the prior role of this node.                               */
  printf("%s %d \n", "  this node's prior role = ", priorRole);

  /* Print which recovery domain this role comes from.                */
  printf("%s", "  Node_Role_Type = ");
  if (crgData->Node_Role_Type == QcstCurrentRcvyDmn)
    printf("%s \n", "QcstCurrentRcvyDmn");
  else
    printf("%s \n", "QcstPreferredRcvyDmn");

  /* Print the ID of the changing node (if any).                      */
  printStr("  Changing_Node_ID = ",
           crgData->Changing_Node_ID, sizeof(Qcst_Node_Id_t));

  /* Print the role of the changing node (if any).                    */
  printf("%s", "  Changing_Node_Role = ");
  if (crgData->Changing_Node_Role == -3)
    printf("%s \n", "*LIST");
  else if (crgData->Changing_Node_Role == -2)
   printf("%s \n", "does not apply");
  else
    printf("%d \n", crgData->Changing_Node_Role);

  /* Print the takeover IP address.                                   */
  printStr("  Takeover_IP_Address = ",
           crgData->Takeover_IP_Address,
sizeof(Qcst_TakeOver_IP_Address_t));

  /* Print the job name.                                              */
  printStr("  Job_Name = ", crgData->Job_Name, 10);

  /* Print the CRG changes.                                           */
```

```
  printf("%s \n", "  Cluster_Resource_Group_Changes = ");
  if (crgData->Cluster_Resource_Group_Changes &
QcstRcvyDomainChange)
    printf("      %s \n", "Recovery domain changed");
  if (crgData->Cluster_Resource_Group_Changes &
QcstTakeOverIpAddrChange)
    printf("      %s \n", "Takeover IP address changed");

  /* Print the failover wait time.                                    */
  printf("%s", "Failover_Wait_Time = ");
  if (crgData->Failover_Wait_Time == QcstFailoverWaitForever)
     printf("%d %s \n", crgData->Failover_Wait_Time, "Wait
forever");
  else if (crgData->Failover_Wait_Time == QcstFailoverNoWait)
     printf("%d %s \n", crgData->Failover_Wait_Time, "No wait");
  else
     printf("%d %s \n", crgData->Failover_Wait_Time, "minutes");

  /* Print the failover default action.                               */
  printf("%s", "Failover_Default_Action = ");
  if (crgData->Failover_Default_Action == QcstFailoverProceed)
     printf("%d %s \n", crgData->Failover_Default_Action,
"Proceed");
  else
     printf("%d %s \n", crgData->Failover_Default_Action,
"Cancel");

  /* Print the failover message queue name.                           */
  printStr("  Failover_Msg_Queue = ",
           crgData->Failover_Msg_Queue,
sizeof(crgData->Failover_Msg_Queue));
  printStr("  Failover_Msg_Queue_Lib = ",
           crgData->Failover_Msg_Queue_Lib,
           sizeof(crgData->Failover_Msg_Queue_Lib));

  /* Print the cluster version.                                       */
  printf("%s %d \n",
         "  Cluster_Version = ", crgData->Cluster_Version);

  /* Print the cluster version mod level                              */
  printf("%s %d \n",
         "  Cluster_Version_Mod_Level = ",
         crgData->Cluster_Version_Mod_Level);

  /* Print the requesting user profile.                               */
  printStr("  Req_User_Profile = ",
           crgData->Req_User_Profile,
sizeof(crgData->Req_User_Profile));

  /* Print the length of the data in the structure.                   */
  printf("%s %d \n",
         "  Length_Info_Returned = ",
crgData->Length_Info_Returned);

  /* Print the offset to the recovery domain array.                   */
  printf("%s %d \n",
         "  Offset_Rcvy_Domain_Array = ",
crgData->Offset_Rcvy_Domain_Array);

  /* Print the number of nodes in the recovery domain array.          */
  printf("%s %d \n",
         "  Number_Nodes_Rcvy_Domain = ",
crgData->Number_Nodes_Rcvy_Domain);

  /* Print the current/new recovery domain.                           */
  printRcvyDomain("   The recovery domain:",
                  crgData->Number_Nodes_Rcvy_Domain,
```

```
                (Qcst_Rcvy_Domain_Array1_t *)
                ((char *)crgData +
crgData->Offset_Rcvy_Domain_Array));

  /* Print the offset to the prior recovery domain array.                  */
  printf("%s %d \n",
         "  Offset_Prior_Rcvy_Domain_Array = ",
         crgData->Offset_Prior_Rcvy_Domain_Array);

  /* Print the number of nodes in the prior recovery domain array.         */
  printf("%s %d \n",
         "  Number_Nodes_Prior_Rcvy_Domain = ",
         crgData->Number_Nodes_Prior_Rcvy_Domain);

  /* Print the prior recovery domain if one was passed.                    */
  if (crgData->Offset_Prior_Rcvy_Domain_Array) {
    printRcvyDomain("   The prior recovery domain:",
                    crgData->Number_Nodes_Prior_Rcvy_Domain,
                    (Qcst_Rcvy_Domain_Array1_t *)
               ((char *)crgData +
crgData->Offset_Prior_Rcvy_Domain_Array));
  }

  return;
}  /* end printParms                                                       */


/*****************************************************************************/
/*                                                                           */
/* Print a string for the action code.                                       */
/*                                                                           */
/*****************************************************************************/
static void printActionCode(unsigned int ac) {

  char *code;
  switch (ac) {
    case QcstCrgAcInitialize: code = "QcstCrgAcInitialize";
                              break;
    case QcstCrgAcStart:      code = "QcstCrgAcStart";
                              break;
    case QcstCrgAcRestart:    code = "QcstCrgAcRestart";
                              break;
    case QcstCrgAcEnd:        code = "QcstCrgAcEnd";
                              break;
    case QcstCrgAcDelete:     code = "QcstCrgAcDelete";
                              break;
    case QcstCrgAcReJoin:     code = "QcstCrgAcReJoin";
                              break;
    case QcstCrgAcFailover:   code = "QcstCrgAcFailover";
                              break;
    case QcstCrgAcSwitchover: code = "QcstCrgAcSwitchover";
                              break;
    case QcstCrgAcAddNode:    code = "QcstCrgAcAddNode";
                              break;
    case QcstCrgAcRemoveNode: code = "QcstCrgAcRemoveNode";
                              break;
    case QcstCrgAcChange:     code = "QcstCrgAcChange";
                              break;
    case QcstCrgAcDeleteCommand: code = "QcstCrgAcDeleteCommand";
                              break;
    case QcstCrgAcUndo:       code = "QcstCrgAcUndo";
                              break;
    case QcstCrgEndNode:      code = "QcstCrgEndNode";
                              break;
    case QcstCrgAcAddDevEnt:  code = "QcstCrgAcAddDevEnt";
                              break;
    case QcstCrgAcRmvDevEnt:  code = "QcstCrgAcRmvDevEnt";
```

```
                              break;
    case QcstCrgAcChgDevEnt:  code = "QcstCrgAcChgDevEnt";
                              break;
    case QcstCrgAcChgNodeStatus: code = "QcstCrgAcChgNodeStatus";
                              break;
    case QcstCrgAcCancelFailover: code = "QcstCrgAcCancelFailover";
                              break;
    case QcstCrgAcVerificationPhase: code =
"QcstCrgAcVerificationPhase";
                              break;
    default:                  code = "unknown action code";
                              break;
  }
  printf("%s", code);

  return;
}  /* end printActionCode                                                  */


/*****************************************************************************/
/*                                                                           */
/* Print the CRG status.                                                     */
/*                                                                           */
/*****************************************************************************/
static void printCrgStatus(int status) {

  char * str;
  switch (status) {
    case QcstCrgActive:               str = "QcstCrgActive";
                                      break;
    case QcstCrgInactive:             str= "QcstCrgInactive";
                                      break;
    case QcstCrgIndoubt:              str = "QcstCrgIndoubt";
                                      break;
    case QcstCrgRestored:             str = "QcstCrgRestored";
                                      break;
    case QcstCrgAddnodePending:       str =
"QcstCrgAddnodePending";
                                      break;
    case QcstCrgDeletePending:        str = "QcstCrgDeletePending";
                                      break;
    case QcstCrgChangePending:        str = "QcstCrgChangePending";
                                      break;
    case QcstCrgEndCrgPending:        str = "QcstCrgEndCrgPending";
                                      break;
    case QcstCrgInitializePending:    str =
"QcstCrgInitializePending";
                                      break;
    case QcstCrgRemovenodePending:    str =
"QcstCrgRemovenodePending";
                                      break;
    case QcstCrgStartCrgPending:      str =
"QcstCrgStartCrgPending";
                                      break;
    case QcstCrgSwitchOverPending:    str =
"QcstCrgSwitchOverPending";
                                      break;
    case QcstCrgDeleteCmdPending:     str =
"QcstCrgDeleteCmdPending";
                                      break;
    case QcstCrgAddDevEntPending:     str =
"QcstCrgAddDevEntPending";
                                      break;
    case QcstCrgRmvDevEntPending:     str =
"QcstCrgRmvDevEntPending";
                                      break;
    case QcstCrgChgDevEntPending:     str =
```

```
"QcstCrgChgDevEntPending";
                                    break;
    case QcstCrgChgNodeStatusPending: str =
"QcstCrgChgNodeStatusPending";
                                    break;
    default: str = "unknown CRG status";
  }
  printf("%s \n", str);

  return;
}  /* end printCrgStatus                                                  */


/*********************************************************************/
/*                                                                   */
/* Print the recovery domain.                                        */
/*                                                                   */
/*********************************************************************/
static void printRcvyDomain(char *str,
                            unsigned int count,
                            Qcst_Rcvy_Domain_Array1_t *rd) {

  unsigned int i;
  printf("\n %s \n", str);
  for (i=1; i&lt;=count; i++) {
    printStr("    Node_ID = ", rd->Node_ID,
sizeof(Qcst_Node_Id_t));
    printf("%s %d \n", "      Node_Role = ", rd->Node_Role);
    printf("%s", "      Membership_Status = ");
    switch (rd->Membership_Status) {
      case 0: str = "Active";
              break;
      case 1: str = "Inactive";
              break;
      case 2: str = "Partition";
              break;
      default: str = "unknown node status";
    }
    printf("%s \n", str);
    rd++;
  }
  return;
}  /* end printRcvyDomain                                               */

/*********************************************************************/
/*                                                                   */
/* Concatenate a null terminated string and a non null terminated string   */
/* and print it.                                                     */
/*                                                                   */
/*********************************************************************/
static void printStr(char *s1, char *s2, unsigned int len) {

  char buffer[132];
  memset(buffer, 0x00, sizeof(buffer));
  memcpy(buffer, s1, strlen(s1));
  strncat(buffer, s2, len);
  printf("%s \n", buffer);
  return;
}  /* end printStr                                                      */
```

Planning data resiliency

Data resilience is the ability for data to be available to users or applications. You can achieve data resiliency by using IBM i cluster technology with either switched disks, cross-site mirroring, or logical replication technologies.

For IBM i supported implementations of data resilience, you have several choices of technologies. When these technologies are combined with IBM i cluster resource services, you can build a complete high-availability solution. These technologies can be categorized this way:

IBM i Independent disk pool technologies

These technologies are all based on IBM i implementation of independent disk pools. For high availability that uses independent disk pool technologies, it is required that all data that needs to be resilient be stored in an independent disk pool. In many cases, this requires migrating data to independent disk pools. This information assumes that migration of data has been completed.

The following IBM i supported technologies are based on independent disk pools:

- Switched disks
- Geographic mirroring
- Metro Mirror
- Global Mirror
- Switched logical units (LUN)

Logical replication technologies

Logical replication is a journal-based technology, where data is replicated to another system in real time. Logical replication technologies use IBM i cluster resource services and journaling with IBM Business Partner applications. These solutions require a high availability business partner application to configure and manage the environment. This information does not provide specific requirements for these IBM Business Partner solutions. If you are implementing a logical replication solution for high availability, consult information related to application or contact a service representative.

Related information:

IBM eServer iSeries Independent ASPs: A Guide to Moving Applications to IASPs

Determine which data should be made resilient

Understand what types of data you should consider making resilient.

Determining which data you need to make resilient is similar to determining which kind of data you need to back up and save when you prepare a back up and recovery strategy for your systems. You need to determine which data in your environment is critical to keeping your business up and running.

For example, if you are running a business on the Web, your critical data can be:

- Today's orders
- Inventory
- Customer records

In general, information that does not change often or that you do not need to use on a daily basis probably does not need to be made resilient.

Planning switched disks

A single copy of the data is maintained on switchable hardware either an expansion unit (tower) or an IOP in a logical partition environment. Tower switching will not be available starting with POWER7 hardware.

When an outage occurs on the primary node, access to the data on the switchable hardware switches to a designated backup node. Additionally, independent disk pools can be used in a cross-site mirroring

(XSM) environment. This allows a mirror copy of the independent disk pool to be maintained on a system that is (optionally) geographically distant from the originating site for availability or protection purposes.

Careful planning is required if you plan to take advantage of switchable resources residing on switchable independent disk pools or cross-site mirroring (XSM).

You should also evaluate your current system disk configuration to determine if additional disk units may be necessary. Similar to any system disk configuration, the number of disk units available to the application can have a significant affect on its performance. Putting additional workload on a limited number of disk units might result in longer disk waits and ultimately longer response times to the application. This is particularly important when it comes to temporary storage in a system configured with independent disk pools. All temporary storage is written to the SYSBAS disk pool. If your application does not use much temporary storage, then you can get by with fewer disk arms in the SYSBAS disk pool. You must also remember that the operating system and basic functions occur in the SYSBAS disk pool.

Before you can use IBM Systems Director Navigator for IBM i to perform any disk management tasks, such as creating an independent disk pool, you need to set up the proper authorizations for dedicated service tools (DST).

Related tasks:

Enabling and accessing disk units

Hardware requirements for switched disks

To use switched disks, you must have specific hardware.

To use switched disks, you must have one of the following:

- One or more expansion units (frame/units) residing on a high-speed link (HSL) loop.
- One or more IOPs on a shared bus or an IOP that is assigned to an I/O pool. In an LPAR environment, you can switch the IOP that contains the independent switched disks between system partitions without having an expansion unit. The IOP must be on the bus shared by multiple partitions or assigned to an I/O pool. All IOAs on the IOP will be switched.

In addition to these hardware requirements the following physical planning is required for switched disks:

- High-speed link (HSL) cables must be used to attach the expansion units to the systems in the cluster. The expansion unit must be physically adjacent in the HSL loop to the alternate system or expansion unit owned by the alternative system. You can include a maximum of two systems (cluster nodes) on each HSL loop, though each system can be connected to multiple HSL loops. You can include a maximum of four expansion units on each HSL loop, though a maximum of three expansion units can be included on each loop segment. On an HSL loop containing two systems, two segments exist, separated by the two systems. All expansion units on one loop segment must be contained in the same device cluster resource group (CRG).
- In order for an expansion unit to become switchable it must physically be the farthest away from the owning system on the loop segment. Note: An error will occur if you try to make an expansion unit switchable if there is another expansion unit farther away the owning system that has not become switchable.
- The switchable expansion unit must be SPCN-cabled to the system unit that will initially serve as the primary node for the device cluster resource group (device CRG). The primary node might be a primary or secondary logical partition within the system unit. If using logical partitions, the system buses in the intended expansion unit must be owned and dedicated by the partition involved in the cluster.

Software requirements for switched disks

If you plan to use switched disks for IBM i high availability, ensure that the minimum software requirements are met.

- To use new and enhanced functions and features of this technology, it is recommended that you install the most current release and version of the operating system on each system or logical partition that is participating in a high-availability solution based on this technology. If the production system and backup system are at different operating system releases, it is required that the backup system be at the more current release.

  Note: For systems on the same HSL loop, see the High Availability Web site to ensure that you have compatible versions of IBM i.
- One of the following graphical interfaces is required to perform some of the disk management tasks necessary to implement independent disk pools.
  - IBM Systems Director Navigator for i
  - System i Navigator
- You need to install IBM i Option 41 HA Switchable Resources. Option 41 gives you the capability to switch independent disk pools between systems. To switch an independent disk pool between systems, the systems must be members of a cluster and the independent switched disk must be associated with a device cluster resource group in that cluster. Option 41 is also required for working with high availability management interfaces which are provided as part of the IBM PowerHA for ilicensed program.

Related information:

High Availability and Clusters

Communications requirements for switched disks

Switched disks require at least one TCP/IP communications interface between the systems in the cluster.

For redundancy, it is recommended that you have at least two separate interfaces between the systems.

Planning cross-site mirroring

Cross-site mirroring provides several i5/OS disaster recovery and high availability technologies: Geographic mirroring, metro mirror, and global mirror.

Cross-site mirroring technologies implement disaster recovery through maintaining separate sites, which are usually at some distance from each other. Each of these technologies have specific communication, hardware, and software requirements. However, before implementing one of these technologies, you should plan your sites as well. One site is typically considered the production or source site. This site contains your production data which is mirrored or copied to the remote site. The remote site, sometime referred to as a backup or target site, contains the mirrored copy of the production data. In the event of site-wide disaster at the production site, the backup site resumes your business with the mirrored data. Before configuring a cross-site mirroring technology, consider the following regarding your site plans.

Determine which sites will be production and backup sites

Access the current hardware and software resources that are in place at each site to determine if there are any missing components that will be necessary for a cross-site mirroring solution.

Determine the distance between production and backup sites

Depending on your communication bandwidth and other factors, distance between sites can affect performance and latency in the mirroring technology you choose. Some cross-site mirroring technologies are better suited for sites that are at great distances, while other may have performance degradation.

Ensure that you have proper authority to DST

Before you can use IBM Systems Director Navigator for i5/OS to perform any disk management tasks you need to set up the proper authorizations for dedicated service tools (DST).

Related tasks:

Enabling and accessing disk units

Planning geographic mirroring

Geographic mirroring is a sub-function of cross-site mirroring. This technology provides disaster recovery and high availability in IBM i environments.

Hardware requirements for geographic mirroring:

If you plan to use geographic mirroring for IBM i high availability, ensure that the minimum hardware requirements are met.

- All independent disk pool hardware requirements must be met.
- At least two IBM i models, which can be separated geographically, are required.
- At least two sets of disks at each site that are roughly of similar capacity are required.
- A separate storage pool for jobs using geographic mirrored independent disk pools should be configured. Performing geographic mirroring from the main storage pool can cause the system to hang under extreme load conditions.
- Geographic mirroring is performed when the disk pool is available. When geographic mirroring is being performed, the system value for the time of day (QTIME) should not be changed.
- Communications requirements for independent disk pools are critical because they affect throughput.
- Geographic mirroring traffic is dispersed round robin across the potentially multiple communication lines available to it. It is recommended that if multiple lines are provided for geographic mirroring, that those lines be of similar speed and capacity.
- It is recommended that a separate communication line be used for the clustering heartbeat to prevent contention with the geographic mirroring traffic.

Related concepts:

"Communications requirements for geographic mirroring" on page 53

When you are implementing an IBM i high-availability solution that uses geographic mirroring, you should plan communication lines so that geographic mirroring traffic does not adversely affect system performance.

Software requirements for geographic mirroring:

If you plan to use geographic mirroring as part of an IBM i high availability solution, the following software is required.

- To use advanced features of geographic mirroring, IBM PowerHA for i license program must be installed.
- To use new and enhanced functions and features of this technology, it is recommended that you install the most current release and version of the operating system on each system or logical partition that is participating in a high-availability solution based on this technology. If the production system and backup system are at different operating system releases, it is required that the backup system be at the more current release.

  Note: For systems on the same HSL loop, see the High Availability Web site to ensure that you have compatible versions of IBM i.
- One of the following graphical interfaces is required to perform some of the disk management tasks necessary to implement independent disk pools.
  - IBM Systems Director Navigator for i
  - System i Navigator
- You need to install IBM i Option 41 HA Switchable Resources. Option 41 gives you the capability to switch independent disk pools between systems. To switch an independent disk pool between systems, the systems must be members of a cluster and the independent switched disk must be associated with

a device cluster resource group in that cluster. Option 41 is also required for working with high availability management interfaces which are provided as part of the IBM PowerHA for ilicensed program.

Related information:

High Availability and Clusters

Communications requirements for geographic mirroring:

When you are implementing an IBM i high-availability solution that uses geographic mirroring, you should plan communication lines so that geographic mirroring traffic does not adversely affect system performance.

The following is recommended:

- Geographic mirroring can generate heavy communications traffic. If geographic mirroring shares the same IP connection with another application, for example clustering, then geographic mirroring might be suspended, which results in synchronization. Likewise, clustering response might be unacceptable, which results in partitioned nodes. Geographic mirroring should have its own dedicated communication lines. Without its own communication line, geographic mirroring can contend with other applications that use the same communication line and affect user network performance and throughput. This also includes the ability to negatively affect cluster heartbeat monitoring, resulting in a cluster partition state. Therefore, it is recommended that you have dedicated communication lines for both geographic mirroring and clusters. Geographic mirroring supports up to four communications lines.

  Geographic mirroring distributes changes over multiple lines for optimal performance. The data is sent on each of the configured communication lines in turn, from 1 to 4, over and over again. Four communication lines allow for the highest performance, but you can obtain relatively good performance with two lines.

  If you use more than one communication line between the nodes for geographic mirroring, it is best to separate those lines into different subnets, so that the usage of those lines is balanced on both systems.
- If your configuration is such that multiple applications or services require the use of the same communication line, some of these problems can be alleviated by implementing Quality of Service (QoS) through the TCP/IP functions of IBM i. The IBM i quality of service (QoS) solution enables the policies to request network priority and bandwidth for TCP/IP applications throughout the network.
- Ensure that throughput for each data port connection matches. This means that the speed and connection type should be the same for all connections between system pairs. If throughput is different, performance will be gated by the slowest connection.
- Consider the delivery method for a geographic mirroring ASP session. Before 7.1, the mirroring uses synchronous communication between the production and mirror copy systems. This delivery method is best for low latency environments. In 7.1, asynchronous support was added, which means asynchronous communications is used between the production and mirror copy systems. This delivery method is best for high latency environments. This delivery method will consume more system resources on the production copy node than synchronous delivery.
- Consider configuring a virtual private network for TCP/IP connections for the following advantages:
  - Security of data transmission by encrypting the data
  - Increased reliability of data transmission by sending greater redundancy

Related concepts:

"Hardware requirements for geographic mirroring" on page 52

If you plan to use geographic mirroring for IBM i high availability, ensure that the minimum hardware requirements are met.

Related reference:

Quality of Service (QoS)

Journal planning for geographic mirroring:

When implementing high availability based on i5/OS geographic mirroring, you should plan for journal management.

Journal management prevents transactions from being lost if your system ends abnormally. When you journal an object, the system keeps a record of the changes you make to that object. Regardless of the high availability solution that you implement, journaling is considered a best practice to prevent data loss during abnormal system outages.

Related information:

Journal management

Backup planning for geographic mirroring:

Before implementing high availability based on geographic mirroring, you should understand and plan a backup strategy within this environment.

Before configuring any high-availability solution, assess your current backup strategy and make appropriate changes if necessary. Geographic mirroring does not allow concurrent access to the mirror copy of the independent disk pool, which has implications to performing remote backups. If you want to back up from the geographically mirrored copy, you must quiesce mirroring on the production system and suspend the mirrored copy with tracking enabled. Tracking allows for changes on the production to be tracked so that they can be synchronized when the mirrored copy comes back online. Then you must vary on the suspended "mirror" copy of the independent disk pool, perform the backup procedure, vary off "the suspended mirror copy" and then resume the independent disk pool to the original production host. This process only requires "partial data resynchronization" between the production and mirrored copies.

Your system is running exposed while doing the backups and when synchronization is occurring. It is also recommended that you suspend mirroring with tracking enabled, which speeds up the synchronization process. Synchronization is also required for any persistent transmission interruption, such as the loss of all communication paths between the source and target systems for an extended period of time. You can also use redundant communication paths to help eliminate some of those risks associated with a communication failure.

It is recommended that you should also use geographic mirroring in at least a three system, or logical partitions, where the production copy of the independent disk pool can be switched to another system at the same site that can maintain geographic mirroring.

Related concepts:

“Scenario: Performing backups in geographic mirroring environment” on page 123
This scenario provides an overview of tasks that are necessary when performing a remote backup in a i5/OS high-availability solution that uses geographic mirroring.

“Scenario: Switched disk with geographic mirroring” on page 83
This scenario describes an i5/OS high-availability solution that uses switched disks with geographic mirroring in a three-node cluster. This solution provides both disaster recovery and high availability.

Performance planning for geographic mirroring:

When implementing a geographic mirroring solution, you need to understand and plan your environment to minimize potential effects on performance.

A variety of factors can influence the performance of geographic mirroring. The following factors provide general planning considerations for maximizing performance in a geographic mirroring environment:

CPU considerations

Geographic mirroring increases the CPU load, so there must be sufficient excess CPU capacity. You might require additional processors to increase CPU capacity. As a general rule, the partitions you are using to run geographic mirroring need more than a partial processor. In a minimal CPU configuration, you can potentially see 5 - 20% CPU overhead while running geographic mirroring. If your backup system has fewer processors in comparison to your production system and there are many write operations, CPU overhead might be noticeable and affect performance.

Base pool size considerations

If asynchronous delivery transmission is used for geographic mirroring, it may be necessary to also increase the amount of storage in the base pool of the system. The amount to increase the base pool by depends primarily on the amount of latency which occurs due to the distance between the two systems. Larger amounts of latency will require larger amounts of the base pool.

Machine pool size considerations

For optimal performance of geographic mirroring, particularly during synchronization, increase your machine pool size by at least the amount given by the following formula:

- The amount of extra machine pool storage is: 300 MB + .3MB x the number of disk ARMs in the independent disk pool. The following examples show the additional machine pool storage needed for independent disk pools with 90 disk ARMs and a 180 disk ARMs, respectively:
  - 300 + (.3 x 90 ARMs) = 327 MB of additional machine pool storage
  - 300 + (.3 x 180 ARMs) = 354 MB of additional machine pool storage

The extra machine pool storage is required on all nodes in the cluster resource group (CRG) so that the target nodes have sufficient storage in case of switchover or failover. As always, the more disk units in the independent disk pool, the better the performance should be, as more things can be done in parallel.

To prevent the performance adjuster function from reducing the machine pool size, you should do one of the following:

1. Set the machine pool minimum size to the calculated amount (the current size plus the extra size for geographic mirroring from the formula) by using Work with Shared Storage Pools (WRKSHRPOOL) command or Change Shared Storage Pool (CHGSHRPOOL) command.

   Note: It is recommended to use this option with the Work with Shared Storage Pools (WRKSHRPOOL) option.
2. Set the Automatically adjust memory pools and activity levels (QPFRADJ) system value to zero, which prohibits the performance adjuster from changing the size of the machine pool.

Disk unit considerations

Disk unit and IOA performance can affect overall geographic mirroring performance. This is especially true when the disk subsystem is slower on the mirrored system. When geographic mirroring is in a synchronous mirroring mode, all write operations on the production copy are gated by the mirrored copy writes to disk. Therefore, a slow target disk subsystem can affect the source-side performance. You can minimize this effect on performance by running geographic mirroring in asynchronous mirroring mode. Running in asynchronous mirroring mode alleviates the wait for the disk subsystem on the target side, and sends confirmation back to the source side when the changed memory page is in memory on the target side.

System disk pool considerations

Similar to any system disk configuration, the number of disk units available to the application can have a significant affect on its performance. Putting additional workload on a limited number of disk units might result in longer disk waits and ultimately longer response times to the application. This is particularly important when it comes to temporary storage in a system configured with independent disk pools. All temporary storage is written to the SYSBAS disk pool. If your application does not use a lot of temporary storage, then you can get by with fewer disk arms in the SYSBAS disk pool. You must also remember that the operating system and basic functions occur in the SYSBAS disk pool. This is also true for the mirror copy system since, in particular, the TCP messages that are sent to the mirror copy can potentially page in the system asp.

Network configuration considerations

Network cabling and configuration can potentially impact geographic mirroring performance. In addition to ensuring that network addressing is set up in different subnets for each set of data port IP addresses, network cabling and configuration should also be set up in the same manner.

Planning metro mirror

i5/OS high availability supports metro mirror, which provides high availability and disaster recovery. To effectively configure and manage a high availability solution that uses this technology, proper planning is required.

Related information:

Guidelines and recommendations for using Copy Services functions with DS6000

Guidelines and recommendations for using Copy Services functions with DS8000

Hardware requirements for metro mirror:

To configure and manage an i5/OS high-availability solution that uses metro mirror technology, you should ensure that the minimum hardware requirements are met.

The following minimum hardware requirements are recommended:

- At least two System i models separated geographically with at least one IBM System Storage® DS8000® external storage unit attached to each system. The DS8000 external storage units are supported on all System i models that support fibre channel attachment for external storage.
- One of the following supported fibre channel adaptors are required:
  - 2766 2 Gigabit Fibre Channel Disk Controller PCI
  - 2787 2 Gigabit Fibre Channel Disk Controller PCI-X
  - 5760 4 Gigabit Fibre Disk Controller PCI-X
- A new IOP is required to support external load source unit on the DS8000:
  - Feature 2847 PCI-X IOP for SAN load source
- Appropriate disk sizing for the system storage should be completed prior to any configuration. You need one set of disk for the source, an equal set of disk units for the target, and another set for each consistency copy.

Related information:

iSeries™ and IBM TotalStorage: A Guide to Implementing External Disk on i5

IBM System Storage DS6000 Information Center

IBM System Storage DS8000 Information Center

Software requirements for Metro Mirror:

Before configuring an IBM i high-availability solution that uses Metro Mirror, ensure that the minimum software requirements have been met.

Metro Mirror has the following minimum software requirements:

- Each IBM i model with in the high-availability solution must be running at least IBM i V6R1 for use with the IBM PowerHA for i licensed program.

  Note: For prior releases, you can still use the IBM Advanced Copy Services for PowerHA on i, which is an offering from Lab Services, to work with IBM System Storage solutions. If you are using Global Mirror on multiple platforms, or if you want to implement Global Mirror on multiple IBM i partitions, you can also use the IBM Advanced Copy Services for PowerHA on i.
- IBM PowerHA for i licensed program installed on each system participating in the high-availability solution that uses Metro Mirror.
- You need to install IBM i Option 41 HA Switchable Resources. Option 41 gives you the capability to switch independent disk pools between systems. To switch an independent disk pool between systems, the systems must be members of a cluster and the independent switched disk must be associated with a device cluster resource group in that cluster. Option 41 is also required for working with high availability management interfaces which are provided as part of the IBM PowerHA for i licensed program.
- To control storage, the IBM PowerHA for i licensed program also requires storage command-line interface (DSCLI). DSCLI is required software for all the IBM System Storage solutions. To manage any of the IBM System Storage solutions, such as the FlashCopy® technology, Metro Mirror, Global Mirror, there is a requirement to have DSCLI installed each of the systems or partitions participating in the high availability solution which uses these storage solutions. DSCLI has these additional software requirements:
  - Java™ Version 1.4
  - Option 35 (CCA Cryptographic Service Provider) installed on each system or partition
- Ensure that the latest PTF have been installed.

Related information:

iSeries™ and IBM TotalStorage: A Guide to Implementing External Disk on i5

IBM System Storage DS6000 Information Center

IBM System Storage DS8000 Information Center

Communications requirement for metro mirror:

Before configuring i5/OS high-availability solution that uses metro mirror, ensure that the minimum communication requirements have been met.

To use the metro mirror technology, you must be using or planning to use a storage area network (SAN).

A *SAN* is a dedicated, centrally managed, secure information infrastructure that enables any-to-any interconnection between systems and storage systems. SAN connectivity is required for using IBM System Storage, such as DS8000 external storage units.

The following are the minimum communication requirements for an i5/OS high availability solution that uses metro mirror:

- One of the following supported fibre channel adaptors are required:
  - 2766 2 Gigabit Fibre Channel Disk Controller PCI
  - 2787 2 Gigabit Fibre Channel Disk Controller PCI-X

  - 5760 4 Gigabit Fibre Disk Controller PCI-X
- The System i product supports a variety of SAN switches and directors. Refer to the Storage area network (SAN) Web site, for a complete list of supported switches and directors
- In addition, taking advantage of multipath I/O is highly recommended in order to improve overall resiliency and performance. Multipath I/O provides the ability to have multiple fibre channel devices configured to the same logical disk units within the storage. When correctly configured this allows single devices, I/O enclosures, or possibly HSL loops to fail without losing connections to the disk units. Multipath also provides performance benefits by spreading workloads across all available connections (paths). Each connection for a multipath disk unit functions independently. Several connections provide improved resiliency by allowing disk storage to be used even if a single path fails.

Related reference:

Storage area network (SAN) Web site

Journal planning for metro mirror:

Journaling is important for increasing recovery time for all high availability solutions. In the case of IBM System Storage based technologies, such as metro mirror, it is vital that journaling be used to force write operations to external storage units, since mirroring of data occurs outside of System i storage.

Journal management prevents transactions from being lost if your system ends abnormally. When you journal an object, the system keeps a record of the changes you make to that object. Regardless of the high availability solution that you implement, journaling is considered a best practice to prevent data loss during abnormal system outages.

Related information:

Journal management

Backup planning for metro mirror:

With metro mirror, you can use the FlashCopy feature to create a copy of data stored in IBM System Storage external storage units.

FlashCopy operations provide the ability to create point-in-time copies. As soon as the FlashCopy operation is processed, both the source and target volumes are available for application use. The FlashCopy feature can be used with other IBM System Storage technologies, such as metro and global mirror, to create consistent, point-in-time copy of data at a remote site which then can be backed up with your standard backup procedures. You should complete the following before implementing the FlashCopy technology:

- Identify the source volumes and target volumes for FlashCopy relationships. You should select FlashCopy target volumes in different ranks for better performance.
- Understand FlashCopy data consistency considerations. There are environments where data is stored in system memory cache and written to disk at some later time. To avoid these types of restart actions, ensure that all data that is related to the FlashCopy source volume has been written to disk before you perform the FlashCopy operation.
- You can use an existing metro mirror source volume as a FlashCopy target volume. This allows you to create a point-in-time copy using a target volume of a FlashCopy pair and then mirror that data to a source metro mirror volume at a remote location.

Performance planning for metro mirror:

You should understand these performance considerations prior to configuring metro mirror.

Before you use metro mirror, consider the following requirements and guidelines:

- The source and target volumes in a metro mirror relationship must be the same storage type.

- The source and target logical volumes must be the same size or the target must be larger in size.
- For metro mirror environments, distribute the work loads by not directing all updates to a small set of common volumes on a single target storage unit. The performance impact at the target site storage unit adversely affects the performance at the source site.
- Similar to any system disk configuration, the number of disk units available to the application can have a significant affect on its performance. Putting additional workload on a limited number of disk units might result in longer disk waits and ultimately longer response times to the application. This is particularly important when it comes to temporary storage in a system configured with independent disk pools. All temporary storage is written to the SYSBAS disk pool. If your application does not use a lot of temporary storage, then you can get by with fewer disk arms in the SYSBAS disk pool. You must also remember that the operating system and basic functions occur in the SYSBAS disk pool.

Related information:

Guidelines and recommendations for using Copy Services functions with DS6000

Guidelines and recommendations for using Copy Services functions with DS8000

Planning global mirror

i5/OS high availability supports global mirror, which provides high availability and disaster recovery in environments that use external storage solutions. To effectively configure and manage high availability that uses this technology, proper planning is required.

IBM System Storage global mirror technology requires all users to share one global mirror connection. i5/OS high availability global mirror allows only one System i™ partition to be active in the global mirror session on a given System Storage server. No other System i partitions or servers from other platforms may use global mirror at the same time. Adding more than one user to a global mirror session will cause unpredictable results to occur.

If you are using global mirror on multiple platforms, or if you want to implement global mirror on multiple System i partitions, you can use the IBM Copy Services for System i. This offering is available from Lab Services.

Related information:

Guidelines and recommendations for using Copy Services functions with DS6000

Guidelines and recommendations for using Copy Services functions with DS8000

Hardware requirements for global mirror:

To configure and manage an i5/OS high-availability solution that uses global mirror technology, you should ensure that the minimum hardware requirements are met.

The following minimum hardware requirements should be met for global mirror:

- At least two System i models separated geographically with at least one IBM System Storage DS8000 external storage unit attached to each system. The DS8000 external storage units are supported on all System i models that support fibre channel attachment for external storage.
- One of the following supported fibre channel adaptors are required:
  - 2766 2 Gigabit Fibre Channel Disk Controller PCI
  - 2787 2 Gigabit Fibre Channel Disk Controller PCI-X
  - 5760 4 Gigabit Fibre Disk Controller PCI-X
- A new IOP is required to support external load source unit on the DS8000:
  - Feature 2847 PCI-X IOP for SAN load source
- Appropriate disk sizing for the system storage should be completed prior to any configuration. You need one set of disk for the source, an equal set of disk units for the target, and another set for each consistency copy.

Related information:

iSeries™ and IBM TotalStorage: A Guide to Implementing External Disk on i5

IBM System Storage DS6000 Information Center

IBM System Storage DS8000 Information Center

Software requirements for Global Mirror:

Before configuring an IBM i high-availability solution that uses Global Mirror, ensure that the minimum software requirements have been met.

Global Mirror has the following minimum software requirements:

- Each IBM i model with in the high-availability solution must be running at least IBM i V6R1 for use with the IBM PowerHA for i licensed program.

  Note: For prior releases, you can still use the IBM Advanced Copy Services for PowerHA on i, which is an offering from Lab Services, to work with IBM System Storage solutions. If you are using Global Mirror on multiple platforms, or if you want to implement Global Mirror on multiple IBM i partitions, you can also use the IBM Advanced Copy Services for PowerHA on i.
- IBM PowerHA for i licensed product install on each system participating in the high-availability solution that use Global Mirror.
- To control storage, the IBM PowerHA for i licensed program also requires storage command-line interface (DSCLI). DSCLI is required software for all the IBM System Storage solutions. To manage any of the IBM System Storage solutions, such as the FlashCopy technology, Metro Mirror, Global Mirror, there is a requirement to have DSCLI installed each of the systems or partitions participating in the high availability solution which uses these storage solutions. DSCLI has these additional software requirements:
  - Java Version 1.4
  - Option 35 (CCA Cryptographic Service Provider) installed on each system or partition
- Ensure that the latest PTF have been installed.

Related information:

iSeries™ and IBM TotalStorage: A Guide to Implementing External Disk on i5

IBM System Storage DS6000 Information Center

IBM System Storage DS8000 Information Center

Communications requirement for global mirror:

Before configuring an i5/OS high-availability solution that uses global mirror, you should ensure that the minimum communication requirements have been met.

To use the global mirror technology you must be using or planning to use a storage area network (SAN).

A *SAN* is a dedicated, centrally managed, secure information infrastructure that enables any-to-any interconnection between systems and storage systems. SAN connectivity is required for using IBM System Storage, such as DS8000 external storage units.

The following are the minimum communication requirements for an i5/OS high availability solution that use global mirror:

- One of the following supported fibre channel adaptors are required:
  - 2766 2 Gigabit Fibre Channel Disk Controller PCI
  - 2787 2 Gigabit Fibre Channel Disk Controller PCI-X

  - 5760 4 Gigabit Fibre Disk Controller PCI-X
- The System i product supports a variety of SAN switches and directors. Refer to the Storage area network (SAN) Web site, for a complete list of supported switches and directors
- In addition, taking advantage of multipath I/O is highly recommended in order to improve overall resiliency and performance. Multipath I/O provides the ability to have multiple fibre channel devices configured to the same logical disk units within the storage. When correctly configured this allows single devices, I/O enclosures, or possibly HSL loops to fail without losing connections to the disk units. Multipath also provides performance benefits by spreading workloads across all available connections (paths). Each connection for a multipath disk unit functions independently. Several connections provide improved resiliency by allowing disk storage to be used even if a single path fails.

Related reference:

Storage area network (SAN) Web site

Journal planning for global mirror:

Journaling is important for decreasing recovery time for all high availability solutions. In the case of IBM System Storage based technologies, such as global mirror, journaling forces write operations to external storage units, which is necessary because data mirroring occurs outside of System i storage.

Journal management prevents transactions from being lost if your system ends abnormally. When you journal an object, the system keeps a record of the changes you make to that object. Regardless of the high availability solution that you implement, journaling is considered a best practice to prevent data loss during abnormal system outages.

Related information:

Journal management

Backup planning for global mirror:

When using global mirror technology within your high-availability solution, you can use the FlashCopy feature to create point-in-time copy of your data.

FlashCopy operations provide the ability to create point-in-time copies. As soon as the FlashCopy operation is processed, both the source and target volumes are available for application use. The FlashCopy feature can be used with other IBM System Storage technologies, such as metro and global mirror, to create consistent, point-in-time copy of data at a remote site which then can be backed up with your standard backup procedures. You should complete the following before implementing the FlashCopy technology:

- Identify the source volumes and target volumes for FlashCopy relationships. You should select FlashCopy target volumes in different ranks for better performance.
- Understand FlashCopy data consistency considerations. There are environments where data is stored in system memory cache and written to disk at some later time. To avoid these types of restart actions, ensure that all data that is related to the FlashCopy source volume has been written to disk before you perform the FlashCopy operation.

Performance planning for global mirror:

You should understand these performance considerations prior to configuring global mirror.

Before you use global mirror, consider these performance guidelines:

- The source and target volumes in a metro mirror relationship must be the same storage type.
- The source and target volumes in a metro mirror relationship must be the same storage type.
- Similar to any system disk configuration, the number of disk units available to the application can have a significant affect on its performance. Putting additional workload on a limited number of disk

units might result in longer disk waits and ultimately longer response times to the application. This is particularly important when it comes to temporary storage in a system configured with independent disk pools. All temporary storage is written to the SYSBAS disk pool. If your application does not use a lot of temporary storage, then you can get by with fewer disk arms in the SYSBAS disk pool. You must also remember that the operating system and basic functions occur in the SYSBAS disk pool.

Related information:

Guidelines and recommendations for using Copy Services functions with DS6000

Guidelines and recommendations for using Copy Services functions with DS8000

Planning for logical replication

Multiple copies of the data are maintained with logical replication. Data is replicated, or copied, from the primary node in the cluster to the backup nodes designated in the recovery domain. When an outage occurs on the primary node, the data remains available because a designated backup node takes over as the primary point of access.

Logical replication makes a copy of something in real time. It is the process of copying objects from one node in a cluster to one or more other nodes in the cluster. Logical replication makes and keeps the objects on your systems identical. If you make a change to an object on one node in a cluster, this change is replicated to other nodes in the cluster.

You must decide on a software technology to use for logical replication. The following solutions are available for achieving logical replication in your cluster:

- **IBM iCluster for i**

  A logical replication product from IBM which provides high availability on IBM i.
- **IBM Business Partners products**

  Data replication software from recognized cluster IBM Business Partners enables you to replicate objects across multiple nodes.
- **A custom-written replication application**

  IBM journal management provides a means by which you can record the activity of objects on your system. You can write an application taking advantage of journal management to achieve logical replication.

Related information:

Journal management

Determine which systems to use for logical replication

When you are determining which systems to use for logical replication, there are several key considerations.

These considerations are:

- Performance capacity
- Disk capacity
- Critical data
- Disaster prevention

If your system fails over, you need to know what data and applications you have running on your primary system and your backup system. You want to put the critical data on the system that is most capable of handling the workload in case it fails over. You do not want to run out of disk space. If your primary system runs out of space and fails over, it is highly likely that your backup system is also going

to fail over due to lack of disk space. To ensure your data center is not completely destroyed in case of a natural disaster, such as a flood, tornado, or hurricane, you should locate the replicated system in a remote location.

Cluster middleware IBM Business Partners and available clustering products

In addition to IBM PowerHA for i, there are other cluster management products available.

IBM iCluster for i, as well as other products, provide software solutions for replication and cluster management functions. Most of these solutions are based on logical replication. Logical replication uses remote journal or similar technology to transfer object changes to a remote system, where they are applied to the target objects. In addition to PowerHA management solutions, you can purchase other cluster middleware products which use logical replication technology. Those products typically also include a management interface.

Journal planning for logical replication

If you are using logical replication, you should use journaling to force writes from the production copy of data to the backup copy of data.

Journal management prevents transactions from being lost if your system ends abnormally. When you journal an object, the system keeps a record of the changes you make to that object. Regardless of the high availability solution that you implement, journaling is considered a best practice to prevent data loss during abnormal system outages.

In logical replication environments, journaling is the basis of the solution and as such is a requirement for implementing a solution based on this technology. With logical replication, a real-time copy to a backup system might be limited depending on the size of the object being replicated. For example, a program updates a record residing within a journaled file. As part of the same operation, it also updates an object, such as a user space, that is not journaled. The backup copy becomes completely consistent when the user space is entirely replicated to the backup system. Practically speaking, if the primary system fails, and the user space object is not yet fully replicated, a manual recovery process is required to reconcile the state of the user space to match the last valid operation whose data was completely replicated.

Related information:

Journal management

Backup planning for logical replication

If you are using a logical replication technology, you should plan for backup operations within this environment.

Logical replication replicates changes to the objects, such as files or programs on a production copy, to a backup copy. The replication is near real time (simultaneous). Typically, if the object, such as a file, is journaled, replication is handled at a record level. A key advantage of this technology is that the backup copy can be accessed in real time for backup operations. You can perform a remote backup on the backup copy of the data without disrupting the production version of the data.

Performance planning for logical replication

If you are using a logical replication technology as part of a high-availability solution, you should understand potential effects on performance with this solution.

With logical replication, potential effects on performance lay in the latency of the replication process. This refers to the amount of lag time between the time at which changes are made on the source system and

the time at which those changes become available on the backup system. Synchronous remote journaling can minimize this to a large extent. Regardless of the transmission mechanism used, you must adequately project your transmission volume and plan your communication lines and speeds correctly to help ensure that your environment can manage replication volumes when they reach their peak. In a high volume environment, latency may be a problem on the target side even if your transmission facilities are properly planned.

Chapter 2. Planning environment resiliency

Environment resiliency ensures that your objects and attributes remain consistent among resources defined in the high-availability environment. You need to identity which resources require a consistent environment to function properly and create a cluster administrative domain that will ensure that these resource attributes remain consistent in your high availability solution.

Planning for a cluster administrative domain

The cluster administrative domain requires planning to manage resources that are synchronized among nodes within a cluster administrative domain. In order to ensure that an application will run consistently on any system in a high-availability environment, all resources that affect the behavior of the application need to be identified, as well as the cluster nodes where the application will run, or where application data might reside.

A cluster administrator can create a cluster administrative domain and add monitored resources that are synchronized among nodes. The i5/OS cluster provides a list of system resources that can be synchronized by a cluster administrative domain, represented by monitored resource entries (MREs).

When designing a cluster administrative domain, you should answer the following questions:

What nodes will be included in the cluster administrative domain?
You should determine what nodes in a cluster are to be managed by the cluster administrative domain. These are the cluster nodes representing the systems where an application can run or where the application data is stored, and that require a consistent operational environment. Nodes cannot be in multiple cluster administrative domains. For example, if you have four nodes in a cluster (Node A, Node B, Node C and Node D), Nodes A and B can be in one cluster administrative domain and Nodes C and D can be in another. However you cannot have Nodes B and C in a third cluster administrative domain and still have them in their original cluster administrative domain.

What will be the naming convention for cluster administrative domains?
Depending on the complexity and size of your clustered environment, you might want to establish a standard naming convention for peer CRGs and cluster administrative domains. Since a peer CRG is created when you create a cluster administrative domain, you will want to differentiate other peer CRGs from those that represent cluster administrative domains. For example, peer CRGs that represent cluster administrative domains can be named *ADMDMN1*, *ADMDMN2*, and so forth, while other peer CRGs can be named *PEER1*. You can also use the List Cluster Resource Group Information (QcstListClusterResourceGroupIn) API to determine whether the peer CRG is used as a cluster administrative domain. A peer CRG which represents a cluster administrative domain can be identified by its application identifier, which is QIBM.AdminDomain.

Planning monitored resources entries (MRE)

Monitored resources are i5/OS objects that can be defined within a cluster administrative domain. These resources need to remain consistent across the systems in a high-availability environment otherwise during an outage applications might not perform as expected. You should plan which supported resources within your environment should be monitored.

You need to determine which system resources need to be synchronized. You can select attributes for each of these resources to customize what is synchronized. Applications that run on multiple nodes might need specific environment variables to run properly. In addition data that spans several nodes might also

require certain user profiles to be accessed. Be aware of the operational requirements for your applications and data before you determine what resources need to be managed by a cluster administrative domain.

Chapter 3. Planning clusters

Before implementing a high-availability solution, you must ensure that you met all prerequisites for clusters.

Hardware requirements for clusters

To implement a high-availability solution, you need to plan and configure a cluster. A cluster groups systems and resources in a high availability environment.

The following are the minimum hardware requirements for clusters:

- You will need at least two System i model or logical partitions. Clusters supports up to 128 systems within a cluster. Any System i model that is capable of running IBM i V4R4M0, or later, is compatible for using clustering.
- External uninterruptible power supply or equivalent is recommended to protect from a sudden power loss which could result in a cluster partition.
- Clustering uses Internet Protocol (IP) multicast capabilities. Multicast does not map well to all types of physical media.
- If you plan to use data resiliency technologies that require independent disk pools, you will also need to plan for hardware specific to your chosen data resiliency technology. You can also use different methods of disk protection to prevent failover from occurring should a protected disk fail.

Related concepts:

"Planning data resiliency" on page 48
Data resilience is the ability for data to be available to users or applications. You can achieve data resiliency by using IBM i cluster technology with either switched disks, cross-site mirroring, or logical replication technologies.

Related reference:

"Planning checklist for clusters" on page 74
Complete the cluster configuration checklist to ensure that your environment is prepared properly before you begin to configure your cluster.

Related information:

Uninterruptible power supply

IP multicasting

Disk protection

Software requirements for clusters

In order to use clustering, you must have the correct software and licenses.

1. Latest supported release of IBM i operating system installed.
2. TCP/IP Connectivity Utilities feature installed.
3. If you plan to use data resiliency technologies, like switched disks or cross-site mirroring, there are additional requirements.
4. Option 41 (High Availability Switchable Resources) is required if you plan to use the following interfaces:
   - IBM PowerHA for i licensed program. This licensed program provides the following interfaces which require Option 41:
     - High Availability Solutions Manager graphical interface
     - Cluster Resource Services graphical interface

- IBM PowerHA for i commands
- IBM PowerHA for i APIs

5. You can also use IBM Business Partner product or write your own high availability management application by using Cluster APIs.

Related concepts:

"Planning switched disks" on page 49
A single copy of the data is maintained on switchable hardware either an expansion unit (tower) or an IOP in a logical partition environment. Tower switching will not be available starting with POWER7 hardware.

"Planning cross-site mirroring" on page 51
Cross-site mirroring provides several i5/OS disaster recovery and high availability technologies: Geographic mirroring, metro mirror, and global mirror.

"Planning data resiliency" on page 48
Data resilience is the ability for data to be available to users or applications. You can achieve data resiliency by using IBM i cluster technology with either switched disks, cross-site mirroring, or logical replication technologies.

Related reference:

"Planning checklist for clusters" on page 74
Complete the cluster configuration checklist to ensure that your environment is prepared properly before you begin to configure your cluster.

Related information:

Cluster APIs

Communications requirements for clusters

Use any type of communications media in your clustering environment as long as it supports Internet Protocol (IP).

Cluster resource services uses TCP/IP and UDP/IP protocols to communicate between nodes. Local area networks (LANs), wide area networks (WANs), OptiConnect system area networks (SANs), or any combination of these connectivity devices are supported. Your choice should be based on the following factors:

- Volume of transactions
- Response time requirements
- Distance between the nodes
- Cost considerations

You can use these same considerations when determining the connection media to be used to connect primary and backup locations of resources. When planning your cluster, it is recommended that you designate one or more of your backup nodes in remote locations in order to survive a site loss disaster.

To avoid performance problems that might be caused by inadequate capacity, you need to evaluate the communication media that is used to handle the volumes of information that are sent from node to node. You can choose which physical media you prefer to use such as token ring, Ethernet, asynchronous transfer mode (ATM), SPD OptiConnect, high-speed lk (HSL) OptiConnect, or Virtual OptiConnect (a high-speed internal connection between logical partitions).

HSL OptiConnect is a technology provided by OptiConnect for IBM i software (IBM i Option 23 - IBM i OptiConnect). It can be used to construct highly available solutions. HSL OptiConnect is a system area network that provides high-speed, point-to-point connectivity between cluster nodes by using high speed link (HSL) Loop technology. HSL OptiConnect requires standard HSL cables, but no additional hardware.

For switchable hardware, also referred to as resilient device CRGs, you need to have an switched disk in your environment. In a logical partition environment, this is a collection of disk units that are on the bus that is being shared by the logical partitions, or that are attached to an input/output processor that has been assigned to an I/O pool. For a multiple system environment, this is one or more switchable expansion units properly configured on the HSL loop also containing the systems in the recovery domain. The switchable expansion unit can also be used in an LPAR environment. .

Note: If you are using 2810 LAN adapters using only TCP/IP, and not using Systems Network Architecture (SNA) or IPX, you can increase your adapter performance on an OS/400® V4R5M0 system by specifying Enable only for TCP(*YES) for your specific line description using the **Work with Line Descriptions (WRKLIND)** command. Enable only for TCP(*YES) is set automatically in OS/400 V5R1M0, and later releases.

Related concepts:

"Planning switched disks" on page 49
A single copy of the data is maintained on switchable hardware either an expansion unit (tower) or an IOP in a logical partition environment. Tower switching will not be available starting with POWER7 hardware.

Related reference:

"Planning checklist for clusters" on page 74
Complete the cluster configuration checklist to ensure that your environment is prepared properly before you begin to configure your cluster.

Dedicate a network for clusters

During normal operations, base clustering communication traffic is minimal. It is, however, highly recommended that you have redundant communication paths configured for each node in a cluster.

A redundant communications path means that you have two lines configured between two nodes in a cluster. If a failure on the first communication path occurs, the second communication path can take over to keep communications running between the nodes, thereby minimizing conditions that can put one or more nodes of the cluster into a cluster partition. One thing to consider when configuring these paths is that if both of your communications lines go into the same adapter on the system, these lines are still at risk if this single adapter fails. However, it should be noted that a cluster partition is not always avoidable. If your system experiences a power loss or if a hardware failure occurs, the cluster can become partitioned. By configuring two lines, you can dedicate one line for clustering traffic and the other line for the normal traffic and also for the backup line if the dedicated line for clustering goes down. The typical network-related cluster partition can best be avoided by configuring redundant communications paths between all nodes in the cluster.

Tips: Cluster communications

Consider these tips when you set up your communications paths.

- Be sure you have adequate bandwidth on your communication lines to handle the non cluster activity along with the clustering heartbeating function and continue to monitor for increased activity.
- For best reliability, do not configure a single communication path linking one or more nodes.
- Do not overburden the line that is responsible for ensuring that you are still communicating with a node.
- Eliminate as many single points of failure as possible, such as having two communication lines coming into a single adapter, same input-output processor (IOP), or same expansion unit.
- If you have an extremely high volume of data being passed over your communication lines, you may want to consider putting data replication and heartbeat monitoring on separate networks.
- User Datagram Protocol (UDP) multicast is the preferred protocol that the cluster communications infrastructure uses to send cluster management information between nodes in a cluster. When the physical media supports multicast capabilities, cluster communications uses the UDP multicast to send

management messaging from a given node to all local cluster nodes that support the same subnet address. Messages that are sent to nodes on remote networks are always sent by using UDP point-to-point capabilities. Cluster communications does not rely on routing capability for multicast messages.

- The multicast traffic that supports cluster management messaging tends to fluctuate by nature. Depending on the number of nodes on a given LAN (that supports a common subnet address) and the complexity of the cluster management structure that is chosen by the cluster administrator, cluster-related multicast packets can easily exceed 40 packets per second. Fluctuations of this nature can have a negative effect on older networking equipment. One example is congestion problems on devices on the LAN that serve as Simple Network Management Protocol (SNMP) agents that need to evaluate every UDP multicast packet. Some of the earlier networking equipment does not have adequate bandwidth to keep up with this type of traffic. You need to ensure that you or the network administrator has reviewed the capacity of the networks to handle UDP multicast traffic to make certain that clustering does not have a negative effect on the performance of the networks.

Performance planning for clusters

Since potentially significant differences exist in your communications environment, you have the capability to adjust variables that affect cluster communications to best match your environment.

The default values should normally be acceptable to most common environments. If your particular environment is not well suited for these defaults, you can tune cluster communications to better match your environment. Base-level tuning and advanced level tuning are available.

Base-level tuning

Base-level tuning allows you to set the tuning parameters to a predefined set of values identified for high, low, and normal timeout and messaging interval values. When the normal level is selected, the default values are used for cluster communications performance and configuration parameters. Selecting the low level causes clustering to increase the heartbeating interval and the various message timeout values. With fewer heartbeats and longer timeout values, the cluster is less sensitive to communications failures. Selecting the high level causes clustering to decrease the heartbeating interval and the various message timeout values. With more frequent heartbeats and shorter timeout values, the cluster is more sensitive to communications failures.

Advanced-level tuning

With advanced-level tuning, individual parameters can be tuned by using a predefined ranges of values. This allows more granular tuning to meet any special circumstances in your communications environment. If an advanced level of tuning is desired, it is recommended that you obtain help from IBM support personnel or equivalent. Setting the individual parameters incorrectly can easily result in decreased performance.

Tunable cluster communications parameters

The Change Cluster Resource Services (QcstChgClusterResourceServices) API enables some of the cluster topology services and cluster communications performance and configuration parameters to be tuned to better suit the many unique application and networking environments in which clustering occurs.

The **Change Cluster (CHGCLU)** command provides a base level of tuning, while the QcstChgClusterResourceServices API provides both base and advanced levels of tuning.

The QcstChgClusterResourceServices API and **Change Cluster Configuration (CHGCLUCFG)** command can be used to tune cluster performance and configuration. The API and command provide a base level of tuning support where the cluster will adjust to a predefined set of values identified for high, low, and normal timeout and messaging interval values. If an advanced level of tuning is desired, usually

anticipated with the help of IBM support personnel, then individual parameters can be tuned through the use of the API over a predefined value range. Inappropriate changes to the individual parameters can easily lead to degraded cluster performance.

When and how to tune cluster parameters

The **CHGCLU** command and the QcstChgClusterResourceServices API provide for a fast path to setting cluster performance and configuration parameters without your needing to understand the details. This base level of tuning primarily affects the heartbeating sensitivity and the cluster message timeout values. The valid values for the base level of tuning support are:

1 (High Timeout Values/Less Frequent Heartbeats)
Adjustments are made to cluster communications to decrease the heartbeating frequency and increase the various message timeout values. With fewer heartbeats and longer timeout values, the cluster will be slower to respond (less sensitive) to communications failures.

2 (Default Values)
Normal default values are used for cluster communications performance and configuration parameters. This setting can be used to return all parameters to the original default values.

3 (Low Timeout Values/More Frequent Heartbeats)
Adjustments are made to cluster communications to decrease the heartbeating interval and decrease the various message timeout values. With more frequent heartbeats and shorter timeout values, the cluster is quicker to respond (more sensitive) to communications failures.

Example response times are shown in the following table for a heartbeat failure leading to a node partition:

Note: Times are specified in minutes:seconds format.

	1 (Less sensitive)			**2 (Default)**			**3 (More sensitive)**		
	Detection of Heartbeat Problem	Analysis	Total	Detection of Heartbeat Problem	Analysis	Total	Detection of Heartbeat Problem	Analysis	Total
Single subnet	00:24	01:02	01:26	00:12	00:30	00:42	00:04	00:14	00:18
Multiple subnets	00:24	08:30	08:54	00:12	04:14	04:26	00:04	02:02	02:06

Depending on typical network loads and specific physical media being used, a cluster administrator might choose to adjust the heartbeating sensitivity and message timeout levels. For example, with a high speed high-reliability transport, such as OptiConnect with all systems in the cluster on a common OptiConnect bus, one might desire to establish a more sensitive environment to ensure quick detection leading to faster failover. Option 3 is chosen. If one were running on a heavily loaded 10 Mbs Ethernet bus and the default settings were leading to occasional partitions just due to network peak loads, option 1 could be chosen to reduce clustering sensitivity to the peak loads.

The Change Cluster Resource Services API also allows for tuning of specific individual parameters where the network environmental requirements present unique situations. For example, consider again a cluster with all nodes common on an OptiConnect bus. Performance of cluster messages can be greatly enhanced by setting the message fragment size parameter to the maximum 32,500 bytes to better match the OptiConnect maximum transmission unit (MTU) size than does the default 1,464 bytes. This reduces the overhead of fragmentation and reassembly of large messages. The benefit, of course, depends on the cluster applications and usage of cluster messaging resulting from those applications. Other parameters are defined in the API documentation and can be used to tune either the performance of cluster

messaging or change the sensitivity of the cluster to partitioning.

Related reference:

QcstChgClusterResourceServices API

Related information:

Change Cluster (CHGCLU) command

Changing cluster resource services settings

The default values affecting message timeout and retry are set to account for most typical installations. However, it is possible to change these values to more closely match your communications environment.

The values can be adjusted in one of these ways:

- Set a general performance level that matches your environment.
- Set values for specific message tuning parameters for more specific adjustment

In the first method, the message traffic is adjusted to one of three communications levels. The normal level is the default and is described in detail in Heartbeat monitoring.

The second method should normally be done only under the advisement of an expert.

The Change Cluster Resource Services (QcstChgClusterResourceServices) API describes details on both methods.

Related reference:

QcstChgClusterResourceServices API

Related information:

Heartbeat monitoring

Planning multiple-release clusters

If you are creating a cluster that includes nodes at multiple cluster versions, then certain steps are required when you create the cluster.

By default, the current cluster version is set to the potential cluster version of the first node added to the cluster. This approach is appropriate if this node is at the lowest version level to be in the cluster. However, if this node is at a later version level, then you cannot add nodes with a lower version level. The alternative is to use the target cluster version value when you create a cluster to set the current cluster version to one less than the potential cluster version of the first node added to the cluster.

Note: If you are using the IBM PowerHA for i licensed program, V6R1 is required on all systems within the cluster.

For example, consider the case where a two-node cluster is to be created. The nodes for this cluster follow:

Node identifier	Release	Potential cluster version
Node A	V5R4	5
Node B	V6R1	6

If the cluster is to be created from Node B, care must be taken to indicate that this will be a mixed-release cluster. The target cluster version must be set to indicate that the nodes of the cluster will communicate at one less than the requesting node's potential node version.

Performance planning for clusters

When changes are made to a cluster, the overhead necessary to manage the cluster can be affected.

The only resources that clustering requires are those necessary to perform heartbeat monitoring, to manage the cluster resource groups and the cluster nodes, and to handle any messaging taking place between cluster resource groups and cluster nodes. After your clustering environment is operational, the only increase in overhead is if you make changes to the cluster.

During a normal operating environment, clustering activity should have minimal effect on your clustered systems.

Planning advanced node failure detection

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

Before implementing advance node failure detection, you must ensure that you have met all the prerequisites.

- To prevent cluster partitions when a cluster node has actually failed, a Hardware Management Console (HMC) v7 or Virtual I/O Server (VIOS) partition can be used.
- For each cluster node, determine the type of server that the cluster monitor will be using to monitor that node. If the node is managed either by an IVM or by an HMC at version V8R8.5.0 or earlier, then a common information model (CIM) server can be used. If the node is managed by an HMC at version V8R8.5.0, then a representational state transfer (REST) server can be used.

  Note: For cluster nodes managed by HMCs at version V8R8.5.0, you may choose to use either a CIM server or a REST server. The REST server is recommended because CIM servers are not supported for later versions of HMC.
- Each node that is to have failures reported to it, will need to have a cluster monitor configured.

Hardware requirements for the advanced node failure detection

Advanced node failure detection feature can be used provided all the hardware requirements are met.

The following minimum hardware requirements are needed for the advanced node failure detection feature:

- At least two IBM i models or logical partitions
- Either Hardware Management Console (HMC) or Virtual I/O Server (VIOS)

Software requirements for the advanced node failure detection

To use the advanced node failure detection function in an IBM i high-availability solution, the minimum software requirements should be met.

Each node planning to use the advanced node failure detection feature with Common Information Model (CIM) servers has the following software requirements:

- 5770-SS1 Base operating system option 33 - Portable Application Solutions Environment
- 5770-SS1 Base operating system option 30 - Qshell
- 5733-SC1 - IBM Portable Utilities for IBM i
- 5733-SC1 option 1 - OpenSSH, OpenSSL, zlib
- 5770-UME IBM Universal Manageability Enablement
- HMC version V8R8.5.0 or earlier. This is the last version of HMC to support the CIM server.
- 5770-HAS IBM PowerHA for i LP

Note: IVM uses the CIM server.

Each node planning to use the advanced node failure detection feature with representational state transfer (REST) servers has the following software requirements:

- 5770-SS1 Base operating system option 3 - Extended Base Directory Support
- 5770-SS1 Base operating system option 33 - Portable Application Solutions Environment
- 5733-SC1 - IBM Portable Utilities for IBM i (Only required for initial configuration of a cluster monitor.)
- 5733-SC1 option 1 - OpenSSH, OpenSSL, zlib (Only required for initial configuration of a cluster monitor.)
- 5770-HAS IBM PowerHA for i LP
- HMC version V8R8.5.0 or later. This is the first version of HMC to support the REST server.
- PowerHA for i new function cluster monitor HMC REST support PTF

Planning checklist for clusters

Complete the cluster configuration checklist to ensure that your environment is prepared properly before you begin to configure your cluster.

Table 1. TCP/IP configuration checklist for clusters

TCP/IP requirements	
__	Start TCP/IP on every node you plan to include in the cluster using the **Start TCP/IP (STRTCP)** Command.
__	Configure the TCP loopback address (127.0.0.1) and verify that it shows a status of Active. Verify the TCP/IP loopback address by using the **Work with TCP/IP Network Status (WRKTCPSTS)** Command on every node in the cluster.
__	Verify that the IP addresses used for clustering on a node have a status of Active. Use the **Work with TCP/IP Network Status (WRKTCPSTS)** Command to check the status of the IP addresses.
__	Verify that the Internet Daemon (INETD) server is active on all nodes in the cluster. If INETD server is not active, you need to start the INETD server. For information about how to start INETD server, see “Starting the INETD server” on page 90.
__	Verify that user profile for INETD, which is specified in /QIBM/ProdData/OS400/INETD/inetd.conf, does not have more than minimal authority. If this user profile has more than minimal authority, starting cluster node will fail. By default, QUSER is specified as the user profile for INETD.
__	Verify every cluster IP address on every node in the cluster can route to and send UDP datagrams to every other IP address in the cluster. If any cluster node uses an IPv4 address, then every node in the cluster needs to have an active IPv4 address (not necessarily configured as a Cluster IP address) that can route to and send TCP packets to that address. Also, if any cluster node uses an IPv6 address, then every node in the cluster needs to have an active IPv6 address (not necessarily configured as a Cluster IP address) that can route to and send TCP packets to that address. Use the **PING** command, specifying a local IP address, and the **TRACEROUTE** command, specifying UDP messages can be useful in determining if two IP addresses can communicate. **PING** and **TRACEROUTE** do not work between IPv4 and IPv6 addresses, or if a firewall is blocking **PING** and **TRACEROUTE**.
__	Verify that ports 5550 and 5551 are not being used by other applications. These ports are reserved for IBM clustering. Port usage can be viewed by using the **Work with TCP/IP Network Status (WRKTCPSTS)** command. Port 5550 is opened and is in a Listen state by clustering after INETD is started.

Table 2. Administrative domain checklist for clusters

Cluster resource services cluster interface considerations	
	Install IBM PowerHA for i (iHASM licensed program (5770-HAS). A valid license key must exist on all cluster nodes that will be in the high-availability solution.
__	Install Option 41 (IBM i - HA Switchable Resources). A valid license key must exist on all cluster nodes that will be in the device domain.

Table 2. Administrative domain checklist for clusters (continued)

Cluster resource services cluster interface considerations	
__	Verify that all host servers are started by using the **Start Host Server (STRHOSTSVR)** Command: STRHOSTSVR SERVER(*ALL)

If you plan to use switchable devices in your cluster, the following requirements must be satisfied:

Table 3. Resilient device configuration checklist for clusters

Resilient device requirements	
__	Install IBM PowerHA for i licensed program. A valid license key must exist on all cluster nodes that will be in the high-availability solution.
__	Verify that Option 41 (HA Switchable Resources) is installed and that a valid license key exists on all cluster nodes that will be in the device domain.
__	To access disk management functions, configure the service tools server (STS) with DST access and user profiles. See Enabling and accessing disk units for details.
__	If you are switching resilient devices between logical partitions on a system, and you are using something other than the HMC to manage your logical partitions, enable Virtual OptiConnect for the partitions. This is done at dedicated service tools (DST) sign-on. See Virtual OptiConnect for details. If you are using the Hardware Management Console to manage your partitions, change your partition profile properties on the OptiConnect tab to enable Virtual OptiConnect for each partition in the switchable configuration. You must activate the partition profile to reflect the change.
__	If an expansion unit on an HSL OptiConnect loop is switched between two systems, and one of the systems has logical partitions, enable HSL OptiConnect for the partitions. If you are using something other than the HMC to manage logical partitions, this is done at dedicated service tools (DST) sign-on. If you are using the Hardware Management Console to manage your partitions, change your partition profile properties on the OptiConnect tab to enable HSL OptiConnect for each partition in the switchable configuration. You must activate the partition profile to reflect the change.
__	If you are switching resilient devices between logical partitions, and you are using something other than the HMC to manage your logical partitions, you must configure the bus to be shared between the partitions or configure an I/O pool. The bus must be configured as Own bus shared by one partition, and all other partitions that will participate in the device switching must be configured as Use bus shared. If you are using the Hardware Management Console to manage your logical partitions, you must configure an I/O pool that includes the I/O processor, I/O adapter, and all attached resources to allow an independent disk pool to be switchable between partitions. Each partition must have access to the I/O pool. See Make your hardware switchable for more details. For details on hardware planning requirements for switchable devices, see Hardware requirements for switched disks.
__	When switching an expansion unit on an HSL loop between two different systems, configure the expansion unit as switchable. See Make your hardware switchable for details.
__	When an expansion unit is added to an existing HSL loop, restart all servers on that same loop.
__	The maximum transmission unit (MTU) for your communication paths must be greater than the cluster communications tunable parameter, Message fragment size. The MTU for a cluster IP address can be verified by using the **Work with TCP/IP Network Status (WRKTCPSTS)** command on the subject node. The MTU must also be verified at each step along the entire communications path. It can be easier to lower the Message fragment size parameter after the cluster is created than raise the MTU for the communications path. See Tunable cluster communications parameters for more information about message fragment size. You can use the Retrieve Cluster Resource Services Information (QcstRetrieveCRSInfo) API to view the current settings of the tuning parameters and the Change Cluster Resource Services (QcstChgClusterResourceServices) API to change the settings.
__	For geographic mirroring, make sure that both nodes are assigned to a different site name.

Table 4. Security configuration checklist for clusters

Security requirements	
__	Set the Allow Add to Cluster (ALWADDCLU) network attribute appropriately on the target node if you are trying to start a remote node. This should be set to *ANY or *RQSAUT depending on your environment. If this attribute is set to *RQSAUT, then IBM i option 34 (Digital Certificate Manager) and the CCA Cryptographic Service Provider (Option 35) must be installed. See Enable a node to be added to a cluster for details on setting the ALWADDCLU network attribute.
__	Enable the status of the user profile for INETD specified in /QIBM/ProdData/OS400/INETD/inetd.conf. It must not have *SECADM or *ALLOBJ special authorities. By default, QUSER is specified as the user profile for INETD.
__	Verify that the user profile that calls the cluster resource services APIs exists on all cluster nodes and has *IOSYSCFG authority.
__	Verify that the user profile to run the exit program for a cluster resource group (CRG) exists on all recovery domain nodes.

Table 5. Job configuration checklist for clusters

Job considerations	
__	Jobs can be submitted by the cluster resource services APIs to process requests. The jobs either run under the user profile to run the exit program specified when creating a cluster resource group, or under the user profile that requested the API (for varying on devices in resilient device CRGs only). Ensure that the subsystem that services the job queue associated with the user profile is configured as: *NOMAX for the number of jobs it can run from that job queue.
__	Jobs are submitted to the job queue specified by the job description that is obtained from the user profile defined for a CRG. The default job description causes the jobs to be sent to the QBATCH job queue. Because this job queue is used for many user jobs, the exit program job might not run in a timely fashion. Consider a unique job description with a unique user queue.
__	When exit program jobs are run, they use routing data from the job description to choose which main storage pool and run time attributes to use. The default values result in jobs that are run in a pool with other batch jobs that have a run priority of 50. Neither of these may produce the desired performance for exit program jobs. The subsystem initiating the exit program jobs (the same subsystem that is using the unique job queue) should assign the exit program jobs to a pool that is not used by other jobs initiated by the same subsystem or other subsystems. In addition, the exit program jobs should be assigned a run priority of 15 so that they run before almost all other user jobs.
__	Set the QMLTTHDACN system value to 1 or 2.

There are several software interfaces available for configuring and managing your cluster. One of these interfaces is Cluster Resource Services interface. If you choose to use Cluster Resource Services, the following requirements must be satisfied.

Table 6. Cluster Resource Services configuration checklist for clusters

Cluster Resource Services graphical interface considerations	
__	Install IBM PowerHA for i licensed program. A valid license key must exist on all cluster nodes that will be in the high-availability solution.
__	Install Option 41 (HA Switchable Resources). A valid license key must exist on all cluster nodes that will be in the device domain.
__	Verify that all host servers are started by using the **Start Host Server (STRHOSTSVR)** command: STRHOSTSVR SERVER(*ALL)

Table 7. Advanced node failure detection checklist for clusters

Advanced node failure detection considerations when using a CIM server or IVM	
__	Determine which cluster nodes are or can be managed with a Hardware Management Console (HMC) or a Virtual I/O Server (VIOS) partition on an Integrated Virtualization Manager (IVM) managed server
__	Determine which cluster node(s) are to receive messages when some other cluster node fails
__	On each cluster node that is to receive a message from an HMC or IVM, the following things must be done.
	Install base operating system option 33 - IBM Portable Application Solutions Environment for i
	Install 5733-SC1 - IBM Portable Utilities for i
	Install 5733-SC1 option 1 - OpenSSH, OpenSSL, zlib
	Install 5770-UME - IBM Universal Manageability Enablement for i.
	Configure the enableAuthentication and sslClientVerificationMode properties for the 5770-UME product.
	Copy a digital certificate file from the HMC or VIOS and add it to an IBM i truststore.
	Start the *CIMOM server with STRTCPSVR *CIMOM CL command
	Configure the cluster monitor(s) with the ADDCLUMON CL command
Advanced node failure detection considerations when using a REST server	
__	Determine which cluster nodes are or can be managed with a Hardware Management Console (HMC) REST server
__	Determine which cluster node(s) are to receive messages when some other cluster node fails
__	On each cluster node that is to receive a message from an HMC, the following things must be done.
	Install base operating system option 3 - Extended Base Directory Support.
	Install base operating system option 33 - IBM Portable Application Solutions Environment for i
	Install 5733-SC1 - IBM Portable Utilities for i (Only required for initial configuration of a cluster monitor.)
	Install 5733-SC1 option 1 - OpenSSH, OpenSSL, zlib (Only required for initial configuration of a cluster monitor.)
	Copy a digital certificate file from the HMC or VIOS and add it to an IBM i truststore.
	Configure the cluster monitor(s) with the ADDCLUMON CL command

Planning the FlashCopy feature

You can use the FlashCopy feature as a means to reduce your backup window in i5/OS high availability environments that use the external storage units of the IBM System Storage. Before using the FlashCopy feature, ensure that the minimum requirements have been met.

Hardware requirements for the FlashCopy feature

To use the FlashCopy technology in an i5/OS high-availability solution, ensure that the minimum hardware requirements are met.

The following minimum hardware requirements are needed for the FlashCopy feature:

- At least two System i models or logical partitions separated geographically with at least one IBM System Storage DS8000 external storage unit attached to each system. The DS8000 external storage units are supported on all System i models that support fibre channel attachment for external storage.
- One of the following supported fibre channel adaptors are required:
  - 2766 2 Gigabit Fibre Channel Disk Controller PCI
  - 2787 2 Gigabit Fibre Channel Disk Controller PCI-X
  - 5760 4 Gigabit Fibre Disk Controller PCI-X

- Appropriate disk sizing for the system storage should be completed prior to any configuration. You need one set of disk for the source, an equal set of disk units for the target, and another set for each consistency copy.

Software requirements for the FlashCopy feature

To use the FlashCopy technology in an IBM i high-availability solution, the minimum software requirements should be met.

The FlashCopy feature has the following minimum software requirements:

- Each IBM i model with in the high-availability solution must be running at least IBM i V6R1 for use with the IBM PowerHA for i licensed program.

  Note: For prior releases, you can still use the IBM Advanced Copy Services for PowerHA on i, which is an offering from Lab Services, to work with IBM System Storage solutions. If you are using Global Mirror on multiple platforms, or if you want to implement Global Mirror on multiple IBM i partitions, you can also use the IBM Advanced Copy Services for PowerHA on i.
- IBM PowerHA for i installed on each system.
- Ensure that the latest PTF have been installed.

Communications requirements for the FlashCopy feature

To use the FlashCopy technology in an i5/OS high-availability solution, ensure that the minimum communications requirements are met.

The following minimum communication requirements should be met for the FlashCopy feature:

- At least two System i models separated geographically with at least one IBM System Storage DS8000 external storage unit attached to each system. The DS8000 external storage units are supported on all System i models that support fibre channel attachment for external storage.
- One of the following supported fibre channel adaptors are required:
  - 2766 2 Gigabit Fibre Channel Disk Controller PCI
  - 2787 2 Gigabit Fibre Channel Disk Controller PCI-X
  - 5760 4 Gigabit Fibre Disk Controller PCI-X
- A new IOP is required to support external load source unit on the DS8000:
  - Feature 2847 PCI-X IOP for SAN load source
- Appropriate disk sizing for the system storage should be completed prior to any configuration. You need one set of disk for the source, an equal set of disk units for the target, and another set for each consistency copy.

Security planning for high availability

Prior to configuring your high-availability solution, you should reassess the current security strategies in your environment and make any appropriate changes to facilitate high availability.

Distributing cluster-wide information

Learn about the security implications of using and managing cluster-wide information.

The Distribute Information (QcstDistributeInformation) API can be used to send messages from one node in a cluster resource group recovery domain to other nodes in that recovery domain. This can be useful in exit program processing. However, it should be noted that there is no encryption of that information. Secure information should not be sent with this mechanism unless you are using a secure network.

Non persistent data can be shared and replicated between cluster nodes by using the Clustered Hash Table APIs. The data is stored in non persistent storage. This means the data is retained only until the

cluster node is no longer part of the clustered hash table. These APIs can only be used from a cluster node that is defined in the clustered hash table domain. The cluster node must be active in the cluster.

Other information distributed by using cluster messaging is similarly not secured. This includes the low level cluster messaging. When changes are made to the exit program data, there is no encryption of the message containing that data.

Considerations for using clusters with firewalls

If you are using clustering in a network that uses firewalls, you should be aware of some limitations and requirements.

If you are using clustering with a firewall, you need to give each node the ability to send outbound messages to and receive inbound messages from other cluster nodes. An opening in the firewall must exist for each cluster address on each node to communicate with every cluster address on every other node. IP packets traveling across a network can be of various types of traffic. Clustering uses ping, which is type ICMP, and also uses UDP and TCP. When you configures a firewall, you can filter traffic based on the type. For clustering to work the firewall needs to allow traffic of ICMP, UDP and TCP. Outbound traffic can be sent on any port and inbound traffic is received on ports 5550 and 5551.

| In addition, if you are making use of advanced node failure detection, any cluster node that is to receive
| failure messages from a Hardware Management Console (HMC) or Virtual I/O Server (VIOS) must be
| able to communicate with that HMC or VIOS. The cluster node will send to the HMC or VIOS on the IP
| address associated with the HMC's or VIOS' domain name and to port 5989. The cluster node will receive
| from the HMC or VIOS on the IP address associated with the cluster node's system name and on port
| 5989.

Maintaining user profiles on all nodes

You can use two mechanisms for maintaining user profiles on all nodes within a cluster.

| In a high-availability environment, a user profile is considered to be the same across systems if the profile
| names are the same. The name is the unique identifier in the cluster. However, a user profile also
| contains a user identification number (UID) and group identification number (GID). To reduce the
| amount of internal processing that occurs during a switchover, where the independent disk pool is made
| unavailable on one system and then made available on a different system, the UID and GID values
| should be synchronized across the recovery domain for the device CRG. Administrative Domain can be
| used to synchronize user profiles, including the UID and GID values, across the cluster.

One mechanism is to create a cluster administrative domain to monitor shared resources across nodes in a cluster. A cluster administrative domain can monitor several types of resources in addition to user profiles, providing easy management of resources that are shared across nodes. When user profiles are updated, changes are propagated automatically to other nodes if the cluster administrative domain is active. If the cluster administrative domain is not active, the changes are propagated after the cluster administrative domain is activated. This method is recommended because it automatically maintains user profiles with a high-availability environment.

With the second mechanism, administrators can also use Management Central in System i Navigator to perform functions across multiple systems and groups of systems. This support includes some common user-administration tasks that operators need to perform across the multiple systems in their cluster. With Management Central, you can perform user profile functions against groups of systems. The administrator can specify a post-propagation command to be run on the target systems when creating a user profile.

Important:

- If you plan to share user profiles that use password synchronization within a cluster, you must set the Retain Server Security (QRETSVRSEC) system value to 1.

- If you change QRETSVRSEC to 0 after you add a monitored resource entry (MRE) for a user profile, and then change a password (if password is being monitored), the global status for the MRE is set to Inconsistent. The MRE is marked as unusable. Any changes made to the user profile after this change are not synchronized. To recover from this problem, change QRETSVRSEC to 1, remove the MRE, and add the MRE again.

Related tasks:

"Creating a cluster administrative domain" on page 110
In a high-availability solution, the cluster administrative domain provides the mechanism that keeps resources synchronized across systems and partitions within a cluster.

Chapter 4. Configuring high availability

Before you can configure a high-availability solution in your i5/OS environment, ensure that you have completed the appropriate planning and understand your resources and goals for high availability and disaster recovery. Use configuration scenarios for high availability and tasks associated with high availability technologies to create your own high-availability solution.

Scenarios: Configuring high availability

Configuration scenarios provide examples of different i5/OS high availability environments and step-by-step configuration tasks to help you implement a high-availability solution based on your needs and resiliency requirements.

These scenarios contain descriptions of the business objectives for high availability and provide a graphic that illustrates the resources within the high-availability solution. Each solution example contains step-by-step instructions to set up and test high availability. However, this information does not cover all configuration cases and additional testing may be required to verify high availability.

Scenario: Switched disk between logical partitions

This scenario describes an i5/OS high-availability solution that uses disk pools that are switched between two logical partitions that reside on a single system.

Overview

Logical partitioning is the ability to make a single i5/OS system function as if it were two or more independent systems. This solution is a good choice for businesses that already have logical partitions configured in their environment.

This scenario does not show the configuration of logical partitions.

Objectives

This solution has the following advantages:

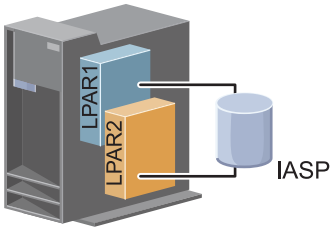
- This is low-cost solution that uses available system resources.
- It provides availability for your business resources during planned outages.
- It provides availability for business resources during some unplanned outages, such as a single logical partition failure.
- Because this solution uses a single copy of data, it minimizes the number of disk units that are required.
- This solution contains current data that does not need to be synchronized.

This solution has the following restrictions:

- There is no disaster recovery available for a site-wide outage.
- There is a requirement that you configure a logical partition.
- There is a possible requirement for redundant hardware between partitions.
- There is only one logical copy of the data that resides in the independent disk pool. This can be a single point of failure, although the data may be protected with RAID protection.
- There is not any concurrent access to the disk pool from both logical partitions.

Details

This graphic illustrates the environment for this scenario:



Configuration steps

Complete the following tasks to configure the high availability technologies associated with this scenario:

1. Complete checklist for cluster
2. Create a cluster
3. Add a node
4. Start a node
5. Add node to a device domain
6. Create a cluster administrative domain
7. Start a cluster administrative domain
8. Create an independent disk pool
9. Add monitored resource entries
10. Make hardware switchable
11. Create a device CRG
12. Starting a device CRG
13. Make disk pool available
14. Perform a switchover to test your high-availability solution

Scenario: Switched disk between systems

This scenario shows an IBM i high-availability solution that uses switched disks between two systems and provides high availability for data, applications, or devices during planned and unplanned outages. There is no switchable tower support between systems on POWER7 hardware.

Overview

By using switched disk technology, this solution provides a simple high-availability solution. With this solution, a single copy of data that is stored in the switched disk always remains current, which eliminates the need to synchronize data among systems and eliminates the risk of losing data during transmission.

Objectives

This solution has the following advantages:

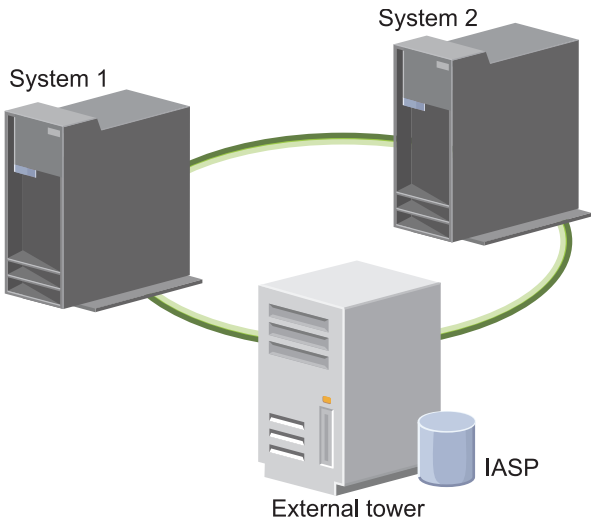
- Provides availability for your business resources during planned outages
- Provides availability for business resources during some unplanned outages
- Enables a single copy of data, which minimizes the number of disk units that are required
- Provides minimal performance overhead
- Enables data to remain current and does not need to be synchronized

This solution has the following restrictions:

- POWER7 hardware will not support switchable towers, so this solution may not be a strategic solution for your business.
- There is not any disaster recovery for a site-wide outage
- There is only one logical copy of the data resides in the independent disk pool. This can be a single point of failure, although the data may be protected with RAID protection.
- There is not any concurrent access to the disk pool from both systems

Details

This graphic illustrates the environment for this scenario:



Configuration steps

1. Complete planning checklist
2. Create a cluster
3. Add a node
4. Start a node
5. Add nodes to device domain
6. Create a cluster administrative domain
7. Start a cluster administrative domain
8. Create an independent disk pool
9. Add monitored resource entries
10. Make hardware switchable
11. Create a device CRG
12. Start a device CRG
13. Make disk pool available
14. Perform a switchover to test your high-availability solution

Scenario: Switched disk with geographic mirroring

This scenario describes an i5/OS high-availability solution that uses switched disks with geographic mirroring in a three-node cluster. This solution provides both disaster recovery and high availability.

Overview

At the production site (Uptown), switched disks are used to move independent disk pools between two nodes. The solution also uses geographic mirroring to generate a copy of the independent disk at a second site (Downtown). Thus, this solution provides both disaster recovery and high availability. The benefits of this solution are essentially the same as the basic switched disk solution with the added advantage of providing disaster recovery to application data by duplicating that data at another location. The production site (Uptown) has an independent disk pool that can be switched between logical partitions to provide high availability with fast switchover times for planned outages, such as applying fixes. This solution also provides disaster recovery with cross-site and geographic mirroring.

Geographic mirroring is a subfunction of cross-site mirroring where data is mirrored to a copy of the independent disk pool at the remote location. Data from the independent disk pool at the production site (Uptown) is mirrored to an independent disk pool on the backup site (Downtown). This solution provides a simple and less expensive alternative to external storage-based solutions, such as IBM System Storage Global Mirror and Metro Mirror. However, geographic mirroring does not offer all the performance options that the external storages solutions provide.

Objectives

This solution has the following advantages:

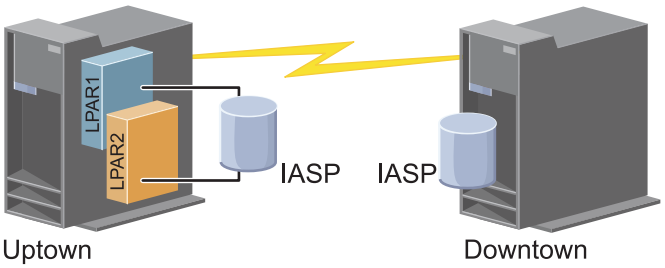
- Provides availability for your business resources during planned outages
- Provides availability for business resources during unplanned outages
- Provides availability for business resources during site-wide disasters
- Enables each site to have a single copy of data which minimizes the number of disk units that are required
- Enables data to remain current and may not need to be synchronized

This solution has the following restrictions:

- There is no concurrent access to the disk pool. However, you can detach the mirror copy for offline processing of a second copy of the data.
- There is potential performance effects with increased central processing unit (CPU) that is required to support geographic mirroring
- Consider using redundant communication paths and adequate bandwidth

Details

This graphic illustrates this solution:



Configuration steps

1. Complete planning checklist for clusters
2. Create a cluster
3. Add a node

4. Start a node
5. Add a node to a device domain
6. Create a device CRG
7. Define site names
8. Create cluster administrative domain
9. Start cluster administrative domain
10. Create an independent disk pool
11. Add monitored resource entries
12. Make hardware switchable
13. Configure geographic mirroring
14. Make disk pools available
15. Perform a switchover to test configuration.

Related tasks:

"Configuring geographic mirroring" on page 119
Geographic mirroring is a sub-function of cross-site mirroring. To configure a high-availability solution by using geographic mirroring, you need to configure a mirroring session between the production system and the backup system.

Scenario: Cross-site mirroring with geographic mirroring

This scenario describes an IBM i high-availability solution that uses geographic mirroring in a two node cluster. This solution provides both disaster recovery and high availability.

Overview

Geographic mirroring is a subfunction of cross-site mirroring where data is mirrored to a copy of the independent disk pool at the remote location. This solution provides disaster recovery in the event of a site-wide outage on the production system (System 1). In that situation, failover to the backup site (System 2) occurs, where operations can continue on the mirrored copy of the data. This solution provides a simple and less expensive alternative to external storage-based solutions, such as IBM System Storage Global Mirror and Metro Mirror. However, geographic mirroring does not offer all the performance options that the external storage solutions provide.

Objectives

This solution has the following advantages:

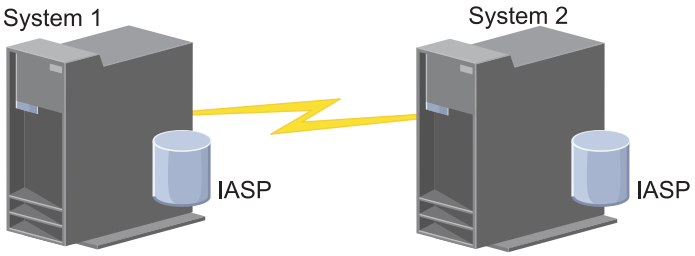
- Provides availability for your business resources during planned outages
- Provides availability for business resources during unplanned outages
- Provides availability for business resources during a disaster
- Enables data to remains current and may not need to be synchronized

This solution has the following restrictions:

- There is no concurrent access to the disk pool. However, you can detach the mirror copy for offline processing of a second copy of the data.
- Potentially affects performance because increased central processing unit (CPU) is required to support geographic mirroring
- Consider using redundant communication paths and adequate bandwidth

Details

The following graphic illustrates this solution:



Configuration steps

1. Complete planning checklist for clusters
2. Create a cluster
3. Add nodes
4. Start nodes
5. Add nodes to device domain
6. Create a cluster administrative domain
7. Start cluster administrative domain
8. Create an independent disk pool
9. Add monitor resource entries
10. Create device CRG
11. Start a device CRG
12. Make disk pool available
13. Configure geographic mirroring.
14. Perform a switchover to test the configuration.

Scenario: Cross-site mirroring with metro mirror

This scenario describes an IBM i high-availability solution which is based on external storage and provides disaster recovery and high availability for storage systems which are separated by short distances. Metro Mirror is an IBM System Storage solution that copies data synchronously from the storage unit at the production site to the storage unit at the backup site. In this way data remains consistent at the backup site.

Overview

The cross-site mirroring with Metro Mirror solution provides a high availability and disaster recovery by using external storage units within a metropolitan area. The independent disk pool is replicated between the external storage devices to provide availability for both planned and unplanned outages. When Metro Mirror receives a host update to the production volume, it completes the corresponding update to the backup volume. Metro Mirror supports a maximum distance of 300 km (186 mi). Delays in response times for Metro Mirror are proportional to the distance between the volumes.

This scenario covers the configuration of IBM-supplied IBM i high availability technology and does not provide installation or configuration instructions regarding IBM System Storage DS8000 series. This information assumes that an IBM System Storage solution is already in place before the i5/OS high availability configuration. For installation and configuration information about DS8000, see IBM System Storage DS8000 Information Center.

Objectives

This solution has the following advantages:

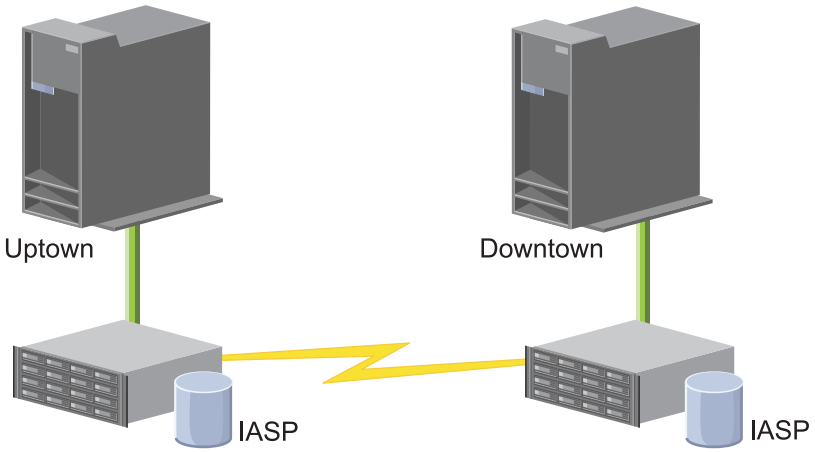
- Replication is entirely managed by the external storage unit, thus no IBM i CPU is used. Replication continues in the storage unit even when the system experiences a system-level failure.
- Availability for business resources during planned or unplanned outages, which includes maintenance outages or software/PTF related outages as well as disaster recovery.
- I/O remains consistent and does not need to be synchronized
- Fast recovery times when used with journaling. Journaling recovers data more quickly in the event of an unplanned outage or failover. Journaling forces data changes to disk where the mirroring is occurring. If you do not use journaling, you might lose data that is in memory. Journaling provides recovery of these data-level transactions and helps with the recovery times.
- The ability to use the FlashCopy function on the source or target side of Metro Mirror.

This solution has the following restrictions:

- Requires external storage hardware
- Consider using redundant communication paths and adequate bandwidth
- There is no concurrent access to the disk pool

Details

The following graphic illustrates this solution:



Configuration steps

1. Complete planning checklist for clusters
2. Create a cluster
3. Add nodes
4. Start nodes
5. Add nodes to device domain
6. Create a cluster administrative domain
7. Start a cluster administrative domain
8. Create an independent disk pool
9. Add monitored resource entries
10. Create a device CRG
11. Start a device CRG

12. Make disk pool available
13. Configure Metro Mirror session
14. Perform a switchover to test the configuration

Scenario: Cross-site mirroring with global mirror

This scenario describes an i5/OS high-availability solution which is based on external storage and provides disaster recovery and high availability for storage systems that are separated by great distances. Global mirror is an IBM Systems Storage solution that copies data asynchronously from the storage unit at the production site to the storage unit at the backup site. In this way, data remains consistent at the backup site.

Overview

The cross-site mirroring with global mirror solution provides a disaster recovery solution by using external storage units across long distances. The independent disk pool is replicated between the external storage devices to provide availability for both planned and unplanned outages.

This scenario covers the configuration of IBM-supplied IBM i high availability technology and does not provide installation or configuration instructions regarding IBM System Storage DS8000 series. This information assumes that an IBM System Storage solution is already in place before the i5/OS high availability configuration. For installation and configuration information about DS8000, see IBM System Storage DS8000 Information Center .

Objectives

Cross-site mirroring with global mirror provides the following advantages:

- Replication is entirely managed by the external storage unit, thus no IBM i CPU is used. Replication continues in the storage unit even when the system experiences a system-level failure.
- Availability for business resources during planned or unplanned outages, which includes maintenance outages or software/PTF related outages as well as disaster recovery.
- Fast recovery times when used with journaling. Journaling recovers data more quickly in the event of an unplanned outage or failover. Journaling forces data changes to disk where the mirroring is occurring. If you do not use journaling, you might lose data that is in memory. Journaling provides recovery of these data-level transactions and helps with the recovery times.
- The ability to use the FlashCopy function on the source or target side of global mirror.

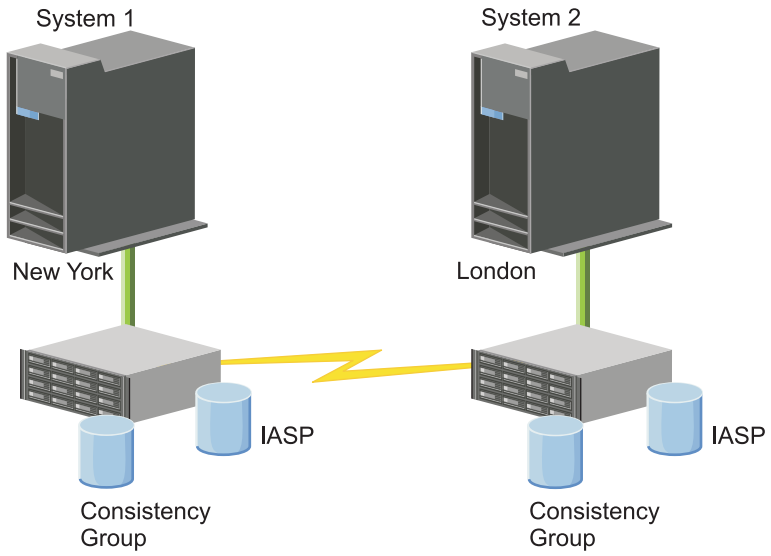
This solution has the following restrictions:

- The solution requires IBM System Storage DS8000 server hardware.
- To achieve acceptable performance, consider using redundant communication paths and adequate bandwidth.
- There is no concurrent access to the disk pool.
- Only one System i partition may configure global mirror on a given System Storage server. No other System i partitions or servers from other platforms may use global mirror at the same time. Adding more than one user to a global mirror session will cause unpredictable results to occur.
- A consistency group is required for the global mirror target copy. A consistency group is not required for the global mirror source copy, but it is highly recommended.
- Reverse replication occurs automatically on a switchover only if the new target has a consistency group. Reverse replication never occurs automatically on a failover.
- When reverse replication does not occur on a switchover or failover, the configuration will consist of two source copies.

- If the desired target copy node has a consistency group, then a reattach operation will convert it to a target copy and automatically initiate replication.
- If the desired target copy node does not have a consistency group, then recovery requires manual intervention with the System Storage DS8000 Storage Manager interface to initiate replication and synchronize the current source and target.

Details

The following graphic illustrates the this solution:



Configuration steps

1. Complete planning checklist for clusters
2. Create a cluster
3. Add nodes
4. Start nodes
5. Add nodes to a device domain
6. Create cluster administrative domain
7. Start a cluster administrative domain
8. Create an independent disk pool
9. Add monitored resource entries
10. Create a device CRG
11. Start a device CRG
12. Make disk pool available
13. Configure global mirror session
14. Perform a switchover to test the configuration

Setting up TCP/IP for high availability

Because cluster resource services uses only IP to communicate with other cluster nodes, which are systems or logical partitions within a high availability environment, all cluster nodes must be IP-reachable, which means that you must have IP interfaces configured to connect the nodes in your cluster.

IP addresses must be set up either manually by the network administrator in the TCP/IP routing tables on each cluster node or they might be generated by routing protocols running on the routers in the network. This TCP/IP routing table is the map that clustering uses to find each node; therefore, each node must have its own unique IP address.

Each node can have up to two IP addresses assigned to it. These addresses must not be changed in any way by other network communications applications. Be sure when you assign each address that you take into account which address uses which kind of communication line. If you have a preference for using a specific type of communication media, make sure that you configure the first IP address by using your preferred media. The first IP address is treated preferentially by the reliable message function and heartbeat monitoring. Every cluster IP address on every node must be able to reach every other IP address in the cluster. If any cluster node uses an IPv4 address, then every node in the cluster needs to have an active IPv4 address (not necessarily configured as a cluster IP address) that can route to and send TCP packets to that address. Also, if any cluster node uses an IPv6 address, then every node in the cluster needs to have an active IPv6 address (not necessarily configured as a cluster IP address) that can route to and send TCP packets to that address. One way to verify that one address can reach another address is if you can ping and use a UDP message trace route in both directions; however, PING and TRACEROUTE do not work between IPv4 and IPv6 addresses, or if a firewall is blocking them.

Note: You need to be sure that the loopback address (127.0.0.1) is active for clustering. This address, which is used to send any messages back to the local node, is normally active by default. However, if it has been ended by mistake, cluster messaging cannot function until this address has been restarted.

Setting TCP/IP configuration attributes

To enable cluster resource services, certain attribute settings are required in the TCP/IP configuration of your network.

You must set these attributes before you can add any node to a cluster:

- Set IP datagram forwarding to *YES by using the **CHGTCPA (Change TCP/IP Attributes)** command if you plan to use a System i product as the router to communicate with other networks and you have no other routing protocols running on that server.
- Set the INETD server to START. See “Starting the INETD server” for information about starting the INETD server.
- Set User Datagram Protocol (UDP) CHECKSUM to *YES using the **CHGTCPA (Change TCP/IP Attributes)** command.
- Set MCAST forwarding to *YES if you are using bridges to connect your token ring networks.
- If you are using OptiConnect for IBM i to communicate between cluster nodes, start the QSOC subsystem by specifying STRSBS(QSOC/QSOC).

Starting the INETD server

The Internet Daemon (INETD) server must be started in order for a node to be added or started, as well as for merge partition processing.

It is recommended that the INETD server always be running in your cluster.

You can start the INETD server through IBM Navigator for i by completing the following steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. In the navigation tree, expand **i5/OS Management** and select **Network**.
4. On the Network page, select **TCP/IP Servers**. A list of available TCP/IP servers is displayed.
5. From the list, select **INETD**.

6. From the **Select Action** menu, select **Start**. The status of the server changes to **Started**.

Alternatively, you can start the INETD server by using the Start TCP/IP Server (STRTCPSVR) command and specifying the SERVER(*INETD) parameter. When the INETD server is started, a User QTCP (QTOGINTD) job is present in the Active Jobs list on the node.

Related reference:

STRTCPSVR (Start TCP/IP Server) command

Configuring clusters

Any i5/OS implementation of high availability requires a configured cluster to control and manage resilient resources. When used with other data resiliency technologies, such as switched disk, cross-site mirroring, or logical replication, cluster technology provides the key infrastructure that is necessary for high-availability solutions.

Cluster resource services provides a set of integrated services that maintain cluster topology, perform heartbeating monitorng, 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 ensures that all nodes have consistent information about the state of cluster resources. The Cluster Resource Service graphical user interface, which is part of the IBM PowerHA for i (iHASM) licensed program number (5770-HAS), allows you to configure and manage clusters within your high-availability solution. In addition, the licensed program also provides a set of control language (CL) commands that will allow you to work with cluster configurations.

There is also application program interfaces (APIs) and facilities that can be used by application providers or customers to enhance their application availability.

In addition to these IBM technologies, high availability business partners provide applications that use clusters with logical replication technology.

Creating a cluster

To create a cluster, you need to include at least one node in the cluster and you must have access to at least one of the nodes that will be in the cluster.

If only one node is specified, it must be the system that you are currently accessing. For a complete list of requirements for creating clusters, see the “Planning checklist for clusters” on page 74.

If you will be using switchable devices in your cluster or by using cross-site mirroring technologies to configure a high-availability solution, there are additional requirements. See Scenarios: Configuring high availability solutions for several configuration examples of high-availability solutions which use these technologies. Each scenario provides step-by-step configuration tasks and an overview of outage coverage this solution provides. You can use these examples to configure your high-availability solution or customize them to suit your needs.

Use the following steps to create a cluster:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Welcome page, select **New Cluster**.
5. Follow the instructions in the New Cluster wizard to create the cluster.

After you create the new cluster, the Welcome page changes to display the name of the cluster at the top of the page. The Welcome page lists several tasks for working with clusters.

After you have created a cluster you need to add any additional nodes and create CRGs.

Related information:

Create Cluster (CRTCLU) command

Create Cluster (QcstCreateCluster) API

Enabling nodes to be added to a cluster

Before you can add a node to a cluster, you need to set a value for the Allow add to cluster (ALWADDCLU) network attribute.

Use the **Change Network Attributes (CHGNETA)** command on any server that you want to set up as a cluster node. The **CHGNETA** command changes the network attributes of a system. The ALWADDCLU network attribute specifies whether a node allows another system to add it as a node in a cluster.

Note: You must have *IOSYSCFG authority to change the network attribute ALWADDCLU.

Possible values follow:

***SAME**
The value does not change. The system is shipped with a value of *NONE.

***NONE**
No other system can add this system as a node in a cluster.

***ANY** Any other system can add this system as a node in a cluster.

***RQSAUT**
Any other system can add this system as a node in a cluster only after the cluster add request has been authenticated.

The ALWADDCLU network attribute is checked to see if the node that is being added is allowed to be part of the cluster and whether to validate the cluster request through the use of X.509 digital certificates. A *digital certificate* is a form of personal identification that can be verified electronically. If validation is required, the requesting node and the node that is being added must have the following installed on the systems:

- IBM i Option 34 (Digital Certificate Manager)
- IBM i Option 35 (CCA Cryptographic Service Provider)

When *RQSAUT is selected for the ALWADDCLU, the certificate authority trust list for the IBM i cluster security server application must be correctly set up. The server application identifier is QIBM_QCST_CLUSTER_SECURITY. At a minimum, add certificate authorities for those nodes that you allow to join the cluster.

Adding nodes

The Cluster Resource Services graphical interface allows you to create a simple two-node cluster when you initially create the cluster. You can add additional nodes to the cluster in your IBM i high availability solution.

If you are creating a new cluster as part of a high availability solution you must add additional nodes through an active node in the cluster.

To add a node to an existing cluster, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the **Cluster Resource Services** page, select the **Work with Cluster Nodes** task to show a list of nodes in the cluster.

5. On the **Nodes** tab, click the **Select Action** menu and select the **Add Node** action. The Add Node page is shown.
6. On the Add Node page, specify the information for the new node. Click **OK** to add the node. The new node appears in the list of nodes. A cluster can contain up to 128 nodes.

Starting nodes

Starting a cluster node starts clustering and cluster resource services on a node in an IBM i high availability environment.

A node can start itself and is able to rejoin the current active cluster, provided it can find an active node in the cluster.

To start clustering on a node, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the **Nodes** tab, select the node you want to start.
5. Click the **Select Action** menu and select **Start**. When cluster resource services is successfully started on the node specified, the status of the node is set to Started.

Adding a node to a device domain

A device domain is a subset of nodes in a cluster that shares device resources.

If you are implementing a high-availability solution that contains independent disk pools-based technologies, such as switched disk or cross-site mirroring, you must define the node as a member of a device domain. After you add the node to a device domain, you can create a device cluster resource group (CRG) that defines the recovery domain for the cluster. All nodes that will be in the recovery domain for a device CRG must be in the same device domain. A cluster node can belong to only one device domain.

To create and manage device domains, you must have i5/OS Option 41 (HA Switchable Resources) installed. A valid license key must exist on all cluster nodes in the device domain.

To add a node to a device domain, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, select **Work with Cluster Nodes** task to show a list of nodes in the cluster.
5. On the **Nodes** tab, select the node that you want to add to the device domain.
6. From the **Select Action** menu, select the **Properties**.
7. On the **Clustering** tab, specify the name of the device domain to which you want to add the node in the **Device domain** field.

Creating cluster resource groups (CRGs)

Cluster resource groups (CRGs) manage high availability resources, such as applications, data, and devices. Each CRG type manages the particular type of resource in a high-availability environment.

The Cluster Resource Services graphical interface allows you to create different CRGs for management of your high availability resources. Each CRG type can be used separately or in conjunction with other CRGs. For example, you may have a stand-alone business application that requires high availability. After you have enabled the application for high availability, you can create CRGs to help manage availability for that application.

If you want only an application, not its data to be available in the event of an outage, you can create an application CRG. However, if you want to have both the data and application available, you can store both within an independent disk pool, which you can define in a device CRG. If an outage occurs, the entire independent disk pool is switched to a backup node, making both the application and its data available.

Creating application CRGs:

If you have applications in your high-availability solution that you want to be highly available, you can create an application cluster resource group (CRG) to manage failovers for that application.

You can specify to allow an active takeover IP address when you create the application CRG. When you start an application CRG that allows for an active takeover IP address, the CRG is allowed to start.

To create an application CRG, complete the following steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, select **Work with Cluster Resource Groups** to show a list of cluster resource groups in the cluster.
5. On the Cluster Resource Group page, click the **Select Action** menu.
6. Select **New Application CRG** and click **Go**. The New Application CRG page is shown:
7. On the **General** page, specify the following information regarding the application CRG:
   - In the **Name** field, specify the name of the CRG. The name cannot exceed 10 characters.
   - In the **Takeover IP address** field, specify the IP address that is to be associated with the application CRG. This value must be in IPv4 or IPv6 format. The takeover IP address allows access to the application without regard to which system the application is currently running on. The **Configure Takeover IP address** field determines whether the user or Cluster Resource Services is responsible for creating the IP address.
   - In the **Description** field, enter a description of the CRG. The description cannot exceed 50 characters.
   - Select **Allow restart** and indicate the number of restart attempts for the application CRG. These values determine the number of attempts to restart the application on the same node before a failover to the backup node occurs.
   - In the **Configure takeover IP address** field, select whether you want Cluster Resource Services or a user to configure and manage takeover IP address for application CRGs. Possible values are:

     Cluster Resource Services
     If you specify this value, the takeover IP address must not exist on any of the nodes in the recovery domain before creating the CRG. It is created for you on all recovery domain nodes. If the IP address exists, then the creation of the application CRG will fail.

     User If you specify this value, you must add the takeover IP address on all primary and backup nodes that are defined in the recovery domain before you can start the CRG.
   - Select **Allow active takeover IP address** to allow a takeover IP address to be active when it is assigned to an application CRG. This field is only valid when the Configure takeover IP address field is set to Cluster Resource Services.
   - In the **Distributed information user queue** field indicate the name of the user queue to receive distributed information. The name cannot exceed 10 characters. In the **Library** field specify the name of the library that contains the user queue to receive the distributed information. The library name cannot be *CURLIB, QTEMP, or *LIBL. The name cannot exceed 10 characters.

     Note: If you set the Distribute information user queue to blank, you must also set the Library name to blank, the Failover wait time to 0, and the Failover default action to 0.

- In the **Failover message queue** field, specify the name of the message queue to receive messages when a failover occurs for this cluster resource group. If this field is set, the specified message queue must exist on all nodes in the recovery domain after the exit program is completed. The failover message queue cannot be in an independent disk pool. In the **Library** field, specify the name of the library that contains the message queue to receive the failover message. The library name cannot be *CURLIB, QTEMP, or *LIBL.
- In the **Failover wait time** field, specify the number of minutes to wait for a reply to the failover message on the cluster message queue. Possible values include:

  Do not wait
  Failover proceeds without user intervention.

  Wait forever
  Failover waits forever until a response is received to the failover inquiry message.

  number
  Specify the number of minutes to wait for a response to the failover inquiry message. If a response is not received in the specified number of minutes, the value in the Failover default action field specifies how to proceed.
- In the **Failover Default Action** field, specify what clustering should do when a response to the failover message on the cluster message queue is not received in the failover wait time limit. You can set this field to **Proceed with failover** or to **Cancel failover**.

8. On the **Exit Program** page, you can specify the information for a CRG exit program. Exit programs are required all CRG types except for device CRGs. Exit programs are called after a cluster-related event for a CRG occurs and responds to that event.
9. On the **Recovery Domain** page, add nodes to the recovery domain and specify their role within the cluster.

Creating data CRGs:

Data cluster resource groups (CRGs) are primarily used with logical replication applications, which are provided by several high availability business partners. If you are implementing a high-availability solution based on logical replication you can create a data CRG to assist the replication of data between primary and backup nodes.

To create a data CRG, complete the following steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, select **Work with Cluster Resource Groups** to show a list of cluster resource groups in the cluster.
5. On the Cluster Resource Group page, click the **Select Action** menu.
6. Select **New Data CRG** and click **Go**. The New Data CRG page displays.
7. On the **General** page, specify the following information regarding the data CRG:
   - In the **Name** field, specify the name of the CRG. The name cannot exceed 10 characters.
   - In the **Description** field, enter a description of the CRG. The description cannot exceed 50 characters.
   - In the **Distributed information user queue** field indicate the name of the user queue to receive distributed information. The name cannot exceed 10 characters. In the **Library** field specify the name of the library that contains the user queue to receive the distributed information. The library name cannot be *CURLIB, QTEMP, or *LIBL. The name cannot exceed 10 characters.

     Note: If you set the Distribute information user queue to blank, you must also set the Library name to blank, the Failover wait time to 0, and the Failover default action to 0.

- In the **Failover message queue** field, specify the name of the message queue to receive messages when a failover occurs for this cluster resource group. If this field is set, the specified message queue must exist on all nodes in the recovery domain after the exit program is completed. The failover message queue cannot be in an independent disk pool. In the **Library** field, specify the name of the library that contains the message queue to receive the failover message. The library name cannot be *CURLIB, QTEMP, or *LIBL.
- In the **Failover wait time** field, specify the number of minutes to wait for a reply to the failover message on the cluster message queue. Possible values include:

  Do not wait
  Failover proceeds without user intervention.

  Wait forever
  Failover waits forever until a response is received to the failover inquiry message.

  number
  Specify the number of minutes to wait for a response to the failover inquiry message. If a response is not received in the specified number of minutes, the value in the Failover default action field specifies how to proceed.

8. On the **Exit Program** page, you can specify the information for a CRG exit program. Exit programs are required all CRG types except for device CRGs. Exit programs are called after a cluster-related event for a CRG occurs and responds to that event.
9. On the **Recovery Domain** page, add nodes to the recovery domain and specify their role within the cluster.

Creating device CRGs:

A device cluster resource group (CRG) is made up of a pool of hardware resources that can be switched as an entity. To create switchable devices within a high-availability solution, the nodes that use these devices need to be a part of a device CRG.

Prior to creating a device CRG, add all nodes that will share a switchable resource to a device domain.

To create a device CRG, complete the following steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, select **Work with Cluster Resource Groups** to show a list of cluster resource groups in the cluster.
5. On the Cluster Resource Group page, click the **Select Action** menu.
6. Select **New Device CRG** and click **Go**. The **New Device CRG** wizard is shown. The **New Device CRG** task is only available if all the nodes in the recovery domain are started.
7. Follow the instructions in the **New Device CRG** wizard to create the new device CRG. While running this wizard, you can create a new device CRG. You can also create either a new independent disk pool or specify an existing disk pool to use.

   The device CRG keeps the hardware resource information identical on all recovery domain nodes and verifies that the resource names are identical. You can also configure a cluster administrative domain to keep the enrolled attributes of the configuration objects, which might include resource names, identical across the cluster administrative domain. If you are using cross-site mirroring, you should create separate device CRGs for independent disk pools and other types of switchable devices at each site.

Creating peer CRGs:

You can create a peer CRG to define node roles in load-balancing environments.

To create a peer CRG in a cluster, complete the following steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, select **Work with Cluster Resource Groups** to show a list of cluster resource groups in the cluster.
5. On the Cluster Resource Group page, click the **Select Action** menu.
6. Select **New Peer CRG** and click **Go**. The New Peer CRG page is shown.
7. On the **General** page, specify the following information regarding the peer CRG:
   - In the **Name** field, specify the name of the CRG. The name cannot exceed 10 characters.
   - In the **Description** field, enter a description of the CRG. The description cannot exceed 50 characters.
   - In the **Application ID** field, specify the application identifier for the peer cluster resource groups in the format *[VendorName].[ApplicationName]*. For example, MyCompany.MyApplication. The identifier cannot exceed 50 characters.
8. On the **Exit Program** page, you can specify the information for a CRG exit program. Exit programs are required all CRG types except for device CRGs. Exit programs are called after a cluster-related event for a CRG occurs and responds to that event.
9. On the **Recovery Domain** page, add nodes to the recovery domain and specify their role within the cluster.

Starting a CRG

Starting a cluster resource group (CRG) activates clustering within your IBM i high availability environment.

To start a CRG, complete the following tasks:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, select **Work with Cluster Resource Groups** to show a list of cluster resource groups in the cluster.
5. On the **Cluster Resource Group** tab, select the name of the CRG that you want to start.
6. From the **Select Action** menu, select **Start**. The Status column shows that the CRG is started.

Related information:

Start Cluster Resource Group (STRCRG) command

Create Cluster Resource Group (QcstCreateClusterResourceGroup) API

Specifying message queues

You can either specify a cluster message queue or a failover message queue. These message queues help you determine causes of failures in your i5/OS high availability environment.

A cluster message queue is used for cluster-level messages and provides one message which controls all cluster resource groups (CRGs) failing over to a specific node. A failover message queue is used for CRG-level messages and provides one message for each CRG that is failing over.

Specifying a cluster message queue

Note: You can also configure a cluster to use a cluster message queue by specifying the message queue while running the Create Cluster wizard.

To specify a cluster message queue, complete these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, click **Display Cluster Properties**.
5. On the Cluster Properties page, click **Cluster Message Queue**.
6. Specify the following information to create a cluster message queue:
   - In the **Name** field, specify the name of the message queue to receive messages that deal with a failover at a cluster or node level. For node-level failovers, one message is sent that controls the failover of all cluster resource groups with the same new primary node. If a cluster resource group is failing over individually, one message is sent that controls the failover of that cluster resource group. The message is sent on the new primary node. If this field is set, the specified message queue must exist on all nodes in the cluster when they are started. The message queue cannot be in an independent disk pool.
   - In the **Library** field, specify the name of the library that contains the message queue to receive the failover message. The library name cannot be *CURLIB, QTEMP, *LIBL, *USRLIBL, *ALL, or *ALLUSR.
   - In **Failover wait time** field, select either **Do not wait** or **Wait forever**, or specify the number of minutes to wait for a reply to the failover message on the cluster message queue.
   - In the **Failover default action** field, specify the action that Cluster Resource Services takes when the response to the failover message has exceeded the failover wait time value. You can set this field to **Proceed with failover** or to **Cancel failover**.

Specifying a failover message queue

To specify a failover message queue, complete these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from your IBM Systems Director Navigator for i5/OS window.
4. On the Cluster Resource Services page, select **Work with Cluster Resource Groups** to show a list of cluster resource groups in the cluster.
5. From the list of cluster resource groups, select the cluster resource group with which you want to work.
6. On the Cluster Resource Group page, click the **Select Action** menu and select **Properties**.
7. On the General page, specify the following values to specify a failover message queue:
   - In the **Failover message queue** field, specify the name of the message queue to receive messages when a failover occurs for this cluster resource group. If this field is set, the specified message queue must exist on all nodes in the recovery domain after the exit program is completed. The failover message queue cannot be in an independent disk pool.
   - In the **Library** field, specify the name of the library that contains the message queue to receive the failover message. The library name cannot be *CURLIB, QTEMP, or *LIBL.
   - In the **Failover wait time** field, specify the number of minutes to wait for a reply to the failover message on the failover message queue. You can also specify the action that Cluster Resource Services takes when a response to the failover message exceeds the specified failover wait time.

Performing switchovers

Switchovers can be performed to test the high availability solution or to handle a planned outages for the primary node, such as a backup operation or scheduled system maintenance.

Performing a manual switchover causes the current primary node to switch over to the backup node. The recovery domain of the cluster resource group defines these roles. When a switchover occurs, the roles of the nodes currently defined in the recovery domain change such that:

- The current primary node is assigned the role of last active backup.
- The current first backup is assigned the role of primary node.
- Subsequent backups are moved up one in the order of backups.

A switchover is only allowed on application, data, and device CRGs that have a status of Active.

Note: If you are performing a switchover on a device CRG, you should synchronize the user profile name, UID, and GID for performance reasons. Cluster administrative domain simplifies synchronization of user profiles.

To perform a switchover on a resource, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, select **Work with Cluster Resource Groups** to show a list of cluster resource groups in the cluster.
5. Select the CRG on which you want to perform a switchover. You can select application CRGs, data CRGs, or device CRGs to perform switchovers.
6. From the **Select Action** menu, select **Switch**.
7. Select **Yes** on the confirmation panel.

The selected cluster resource group is now switched to the backup node. The Status column is updated with the new node name.

Related concepts:

Cluster administrative domain

Related tasks:

"Configuring cluster administrative domains" on page 110
In a high-availability environment, it is necessary that the application and operational environment remain consistent among the nodes that participate in high availability. Cluster administrative domain is the i5/OS implementation of environment resiliency and ensures that the operational environment remains the consistent across nodes.

Related information:

Change Cluster Resource Group Primary (CHGCRGPRI) command

Initiate Switchover (QcstInitiateSwitchOver) API

Configuring nodes

Nodes are systems or logical partitions that are participating in an i5/OS high availability solution.

There are several tasks related to node configuration. When you use the Create Cluster wizard, you can configure a simple two-node cluster. You can add can add additional nodes up to a total of 128. Depending on the technologies that comprise your high-availability solution, additional node configuration tasks might be required.

Starting nodes

Starting a cluster node starts clustering and cluster resource services on a node in an IBM i high availability environment.

A node can start itself and is able to rejoin the current active cluster, provided it can find an active node in the cluster.

To start clustering on a node, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the **Nodes** tab, select the node you want to start.
5. Click the **Select Action** menu and select **Start**. When cluster resource services is successfully started on the node specified, the status of the node is set to Started.

Related information:

Start Cluster Node (STRCLUNOD)) command

Start Cluster Node (QcstStartClusterNode) API

Enabling nodes to be added to a cluster

Before you can add a node to a cluster, you need to set a value for the Allow add to cluster (ALWADDCLU) network attribute.

Use the **Change Network Attributes (CHGNETA)** command on any server that you want to set up as a cluster node. The **CHGNETA** command changes the network attributes of a system. The ALWADDCLU network attribute specifies whether a node allows another system to add it as a node in a cluster.

Note: You must have *IOSYSCFG authority to change the network attribute ALWADDCLU.

Possible values follow:

***SAME**
The value does not change. The system is shipped with a value of *NONE.

***NONE**
No other system can add this system as a node in a cluster.

***ANY** Any other system can add this system as a node in a cluster.

***RQSAUT**
Any other system can add this system as a node in a cluster only after the cluster add request has been authenticated.

The ALWADDCLU network attribute is checked to see if the node that is being added is allowed to be part of the cluster and whether to validate the cluster request through the use of X.509 digital certificates. A *digital certificate* is a form of personal identification that can be verified electronically. If validation is required, the requesting node and the node that is being added must have the following installed on the systems:

- IBM i Option 34 (Digital Certificate Manager)
- IBM i Option 35 (CCA Cryptographic Service Provider)

When *RQSAUT is selected for the ALWADDCLU, the certificate authority trust list for the IBM i cluster security server application must be correctly set up. The server application identifier is QIBM_QCST_CLUSTER_SECURITY. At a minimum, add certificate authorities for those nodes that you allow to join the cluster.

Adding nodes

The Cluster Resource Services graphical interface allows you to create a simple two-node cluster when you initially create the cluster. You can add additional nodes to the cluster in your IBM i high availability solution.

If you are creating a new cluster as part of a high availability solution you must add additional nodes through an active node in the cluster.

To add a node to an existing cluster, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the **Cluster Resource Services** page, select the **Work with Cluster Nodes** task to show a list of nodes in the cluster.
5. On the **Nodes** tab, click the **Select Action** menu and select the **Add Node** action. The Add Node page is shown.
6. On the Add Node page, specify the information for the new node. Click **OK** to add the node. The new node appears in the list of nodes. A cluster can contain up to 128 nodes.

Related information:

Add Cluster Node Entry (ADDCLUNODE) command

Add Cluster Node Entry (QcstAddClusterNodeEntry) API

Adding a node to a device domain

A device domain is a subset of nodes in a cluster that shares device resources.

If you are implementing a high-availability solution that contains independent disk pools-based technologies, such as switched disk or cross-site mirroring, you must define the node as a member of a device domain. After you add the node to a device domain, you can create a device cluster resource group (CRG) that defines the recovery domain for the cluster. All nodes that will be in the recovery domain for a device CRG must be in the same device domain. A cluster node can belong to only one device domain.

To create and manage device domains, you must have i5/OS Option 41 (HA Switchable Resources) installed. A valid license key must exist on all cluster nodes in the device domain.

To add a node to a device domain, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, select **Work with Cluster Nodes** task to show a list of nodes in the cluster.
5. On the **Nodes** tab, select the node that you want to add to the device domain.
6. From the **Select Action** menu, select the **Properties**.
7. On the **Clustering** tab, specify the name of the device domain to which you want to add the node in the **Device domain** field.

Related information:

Add Device Domain Entry (ADDDEVDMNE) command

Add Device Domain Entry (QcstAddDeviceDomainEntry) API

Configuring advanced node failure detection

Advanced node failure detection can be used to prevent cluster partitions when a cluster node has actually failed. A Hardware Management Console (HMC) or Virtual I/O Server (VIOS) partition can be used.

In this example, a 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.

For advanced node failure detection, follow these steps:

1. Configure the HMC with CIM server, HMC with REST server or VIOS . If you are using HMC with the REST server, skip to step 10.
2. The *CIMOM TCP server must be configured and started on each cluster node that has a cluster monitor configured on it. The default configuration of the *CIMOM server that is provided by the installation of the 5770-UME LP must be changed so that the IBM i system can communicate with the CIM server. In order to do that, two configuration attributes that control security aspects need to be changed by running the **cimconfig** command within a PASE shell.
3. Start the server from the command line with **STRTCPSVR *CIMOM**
4. Start a PASE shell from the command line with **CALL QP2TERM**
5. Enter **/QOpenSys/QIBM/ProdData/UME/Pegasus/bin/cimconfig -s enableAuthentication=false -p**
   See Authentication on CIMON for more information about enableAuthentication attribute.

6. Enter **/QOpenSys/QIBM/ProdData/UME/Pegasus/bin/cimconfig -s sslClientVerificationMode=optional -p** See Authentication on CIMOM for more information about sslClientVerificationMode attribute.
7. End the PASE shell by pressing **F3**.
8. End the *CIMOM server with **ENDTCPSVR *CIMOM** .
9. Restart the *CIMOM server from the command line with **STRTCPSVR *CIMOM**.
10. A digital certificate file from the HMC or VIOS partition must be copied to the cluster node and added to a truststore. The digital certificates are self signed by the HMC or VIOS partition. Installing a new version of software on the HMC or VIOS partition may generate a new certificate which will then cause communication between the HMC or VIOS partition and the cluster node to fail (you will see error CPFBBCB with error code 4). If this occurs, add the digital certificate to the truststore on the nodes which have that HMC or VIOS partition configured in a cluster monitor.
11. To perform the cluster configuration steps, you may use either use the command line interface, **Add Cluster Monitor (ADDCLUMON)** command or a web browser. The **Add Cluster Monitor** command, must be used if you want to use the representational state transfer (REST) server. The PowerHA graphical interface only supports the Common Informational Model (CIM) server for the cluster monitor. Should you choose the latter, perform the steps below:
    a. Enter ***http://mysystem:2001***, where ***mysystem*** is the host name of the system.
    b. Log on to the system with your user profile and password.
    a. Select **PowerHA** from the IBM System Director Navigator for i window.
    b. Select **Work with Cluster Node**s.
    c. Select the pop-up menu for a node.
    d. Select **Properties**.
    e. Select **Monitors**.
    f. Select Action: **Add Cluster Monitor.**
    g. Enter correct CIM server host name, user ID, and password.
    h. Press OK.

Configuring advanced node failure detection on hardware management console (HMC) with CIM server

A Hardware Management Console (HMC) can be used with advance node failure detection to prevent cluster partitions when a cluster node has actually failed.

For HMC setup, follow these steps:

1. Follow these steps when your HMC level is V8R8.5.0 or less.
2. Ensure the *CIMOM TCP server is running on your IBM i. You can look for the QUMECIMOM job within the QSYSWRK subsystem to see whether it is running. If the job is not running, you can start it with the command STRTCPSVR *CIMOM
3. Ensure the *SSHD TCP server is running on your IBM i. (on the green screen command entry display: STRTCPSVR *SSHD). In order to start the *SSHD server, you need to ensure that the QSHRMEMC system value is set to 1.
4. You must use the physical monitor and keyboard attached to your HMC. You cannot telnet or use a web interface to the HMC
5. Open a restricted shell by right-clicking on the desktop, then select terminals/xterm.
6. You will get a new shell window on the desk top in which you can enter commands.
7. In the next step you, will be using the secure copy command on the HMC. However, you must have a home directory associated with your profile on the IBM i. For example, if you use QSECOFR as the profile name on the **scp** command, you will need to have a /home/QSECOFR directory created in the integrated file system on the IBM i.

8. Use the secure copy command to copy a file to your IBM i cluster node. (scp /etc/Pegasus/server.pem QSECOFR@LP0236A:/server_name.pem) In the above command, change LP0236A to the name of your IBM i system name and change the server_name.pem to hmc_name.pem. For example, name the file myhmc.pem.
9. Sign off the HMC
10. Sign on your IBM i system and bring up a green screen command entry display
11. Enter the PASE shell environment. (on the green screen command entry display: call qp2term)
12. Move the HMC digital certificate (mv /myhmc.pem /QOpenSys/QIBM/UserData/UME/Pegasus/ssl/truststore/myhmc.pem (in the above, replace the name, myhmc.pem, with your specific file name)
13. Add the digital certificate to the truststore (/QOpenSys/QIBM/ProdData/UME/Pegasus/bin/cimtrust -a -U QSECOFR -f /QOpenSys/QIBM/UserData/UME/Pegasus/ssl/truststore/myhmc.pem -T s)
14. In the above, replace the name, myhmc.pem, with your specific file name.
15. Exit the PASE shell by pressing F3.
16. End the CIM server. On the green screen command entry display: **ENDTCPSVR *CIMOM**.
17. Restart the CIM server to pick up the new certificate. (on the green screen command entry display: **STRTCPSVR *CIMOM**

Configuring advanced node failure detection on hardware management console (HMC) with REST server

A Hardware Management Console (HMC) can be used with advance node failure detection to prevent cluster partitions when a cluster node has actually failed.

HMC with a representational state transfer (REST) server requires HMC version V8R8.5.0 or greater.

For HMC setup, follow these steps:

1. Enter the PASE shell environment. On the green screen command entry display enter `call qp2term`.
2. Get the HMC certificate, by issuing

   ```
   echo -n | openssl s_client -connect hmc-name:443 | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > hmc-name.cert
   ```

   Replace `hmc-name` with the actual name of your HMC. This copies the certificate into a file named `hmc-name.cert` in your current directory.
3. In a web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
4. Log on to the system with your user profile and password.
5. Click **Internet Configurations** from the IBM Navigator for i window.
6. On the **Internet Configurations** page, click **Digital Certificate Manager**. You need to enter your user profile and password on the window that pops up.
7. On the **Digital Certificate Manager** page, click **Create New Certificate Store**.
8. On the page that appears, you should have an option for ***SYSTEM**. Make sure that the button is selected and click **Continue**.

   Note: If the *SYSTEM option is not there, you already have a *SYSTEM store created. Skip forward to step 11.
9. Select **No - Do not create a certificate in the certificate store**.
10. Create a password for the *SYSTEM store and click **Continue**. The password is case-sensitive. Most special characters work (sometimes there are problems with the @ symbol). It is recommended that you do not use special characters. The password is not tied to a user profile, so it will not lock you out of the system after too many retries if you forget it.

11. You have successfully create the *SYSTEM store. Now that the *SYSTEM store is created, you need to click **OK** and then click **Select a Certificate Store**. Select the newly created *SYSTEM option, and click **continue**.
12. Sign in with your new password and click **Continue**.
13. Click **Manage Certificates**.
14. Select **Import certificate** and click **Continue**.
15. Select **Certificate Authority (CA)** and click **Continue**.
16. Enter the path name of the certificate you want to import. For example, the path and file name may be `/hmc-name.cert`. Click **Continue**.

Configuring virtual I/O server (VIOS)

A Virtual I/O Server (VIOS) can be used with advance node failure detection to prevent cluster partitions when a cluster node has actually failed.

The representational state transfer (REST) server is not supported by IVM.

For a VIOS partition, follow these steps:

1. Ensure the *SSHD TCP server is running on your IBM i. On the green screen command line enter: **STRTCPSVR *SSHD**.
2. Telnet to and sign on to the VIOS partition.
3. Change to a non restricted shell by entering oem_setup_env
4. Use the secure copy command to copy a file to your IBM i cluster node. For example, /usr/bin/scp /opt/freeware/cimom/pegasus/etc/cert.pem QSECOFR@system-name:/server.pem. Change **system-name** to the IBM i system name. Change **server.pem** to vios-name.pem.
5. Start the CIMOM server running in the VIOS partition by entering **startnetsvc cimserver**.
6. Sign off the VIOS partition.
7. On the IBM i system, sign on to a green screen command line.
8. Enter the PASE shell environment. On the green screen command line, enter **call qp2term**.
9. Move the HMC digital certificate, enter mv **/vios1.pem /QOpenSys/QIBM/UserData/UME/Pegasus/ssl/truststore/vios1.pem**. Replace vios1.pem, with your specific file name.
10. Add the digital certificate to the truststore, enter **/QOpenSys/QIBM/ProdData/UME/Pegasus/bin/cimtrust -a -U QSECOFR -f vios1.pem -T s**. Replace the vios1.pem name, with your specific file name.
11. Exit the PASE shell by pressing **F3**.
12. End the CIMOM server. On the green screen command line, enter **ENDTCPSVR *CIMOM**.
13. Restart the CIMOM server to pick up the new certificate. On the green screen command line, enter S**TRTCPSVR *CIMOM**.

Configuring CRGs

Cluster resource groups (CRGs) manage resources within an i5/OS high availability environment. Several task enable the management of high availability resources through CRGs.

Starting a CRG

Starting a cluster resource group (CRG) activates clustering within your IBM i high availability environment.

To start a CRG, complete the following tasks:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.

4. On the Cluster Resource Services page, select **Work with Cluster Resource Groups** to show a list of cluster resource groups in the cluster.
5. On the **Cluster Resource Group** tab, select the name of the CRG that you want to start.
6. From the **Select Action** menu, select **Start**. The Status column shows that the CRG is started.

Related information:

Start Cluster Resource Group (STRCRG) command

Create Cluster Resource Group (QcstCreateClusterResourceGroup) API

Creating cluster resource groups (CRGs)

Cluster resource groups (CRGs) manage high availability resources, such as applications, data, and devices. Each CRG type manages the particular type of resource in a high-availability environment.

The Cluster Resource Services graphical interface allows you to create different CRGs for management of your high availability resources. Each CRG type can be used separately or in conjunction with other CRGs. For example, you may have a stand-alone business application that requires high availability. After you have enabled the application for high availability, you can create CRGs to help manage availability for that application.

If you want only an application, not its data to be available in the event of an outage, you can create an application CRG. However, if you want to have both the data and application available, you can store both within an independent disk pool, which you can define in a device CRG. If an outage occurs, the entire independent disk pool is switched to a backup node, making both the application and its data available.

Creating application CRGs:

If you have applications in your high-availability solution that you want to be highly available, you can create an application cluster resource group (CRG) to manage failovers for that application.

You can specify to allow an active takeover IP address when you create the application CRG. When you start an application CRG that allows for an active takeover IP address, the CRG is allowed to start.

To create an application CRG, complete the following steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, select **Work with Cluster Resource Groups** to show a list of cluster resource groups in the cluster.
5. On the Cluster Resource Group page, click the **Select Action** menu.
6. Select **New Application CRG** and click **Go**. The New Application CRG page is shown:
7. On the **General** page, specify the following information regarding the application CRG:
   - In the **Name** field, specify the name of the CRG. The name cannot exceed 10 characters.
   - In the **Takeover IP address** field, specify the IP address that is to be associated with the application CRG. This value must be in IPv4 or IPv6 format. The takeover IP address allows access to the application without regard to which system the application is currently running on. The **Configure Takeover IP address** field determines whether the user or Cluster Resource Services is responsible for creating the IP address.
   - In the **Description** field, enter a description of the CRG. The description cannot exceed 50 characters.
   - Select **Allow restart** and indicate the number of restart attempts for the application CRG. These values determine the number of attempts to restart the application on the same node before a failover to the backup node occurs.

- In the **Configure takeover IP address** field, select whether you want Cluster Resource Services or a user to configure and manage takeover IP address for application CRGs. Possible values are:

  Cluster Resource Services
  If you specify this value, the takeover IP address must not exist on any of the nodes in the recovery domain before creating the CRG. It is created for you on all recovery domain nodes. If the IP address exists, then the creation of the application CRG will fail.

  User If you specify this value, you must add the takeover IP address on all primary and backup nodes that are defined in the recovery domain before you can start the CRG.
- Select **Allow active takeover IP address** to allow a takeover IP address to be active when it is assigned to an application CRG. This field is only valid when the Configure takeover IP address field is set to Cluster Resource Services.
- In the **Distributed information user queue** field indicate the name of the user queue to receive distributed information. The name cannot exceed 10 characters. In the **Library** field specify the name of the library that contains the user queue to receive the distributed information. The library name cannot be *CURLIB, QTEMP, or *LIBL. The name cannot exceed 10 characters.

  Note: If you set the Distribute information user queue to blank, you must also set the Library name to blank, the Failover wait time to 0, and the Failover default action to 0.
- In the **Failover message queue** field, specify the name of the message queue to receive messages when a failover occurs for this cluster resource group. If this field is set, the specified message queue must exist on all nodes in the recovery domain after the exit program is completed. The failover message queue cannot be in an independent disk pool. In the **Library** field, specify the name of the library that contains the message queue to receive the failover message. The library name cannot be *CURLIB, QTEMP, or *LIBL.
- In the **Failover wait time** field, specify the number of minutes to wait for a reply to the failover message on the cluster message queue. Possible values include:

  Do not wait
  Failover proceeds without user intervention.

  Wait forever
  Failover waits forever until a response is received to the failover inquiry message.

  number
  Specify the number of minutes to wait for a response to the failover inquiry message. If a response is not received in the specified number of minutes, the value in the Failover default action field specifies how to proceed.
- In the **Failover Default Action** field, specify what clustering should do when a response to the failover message on the cluster message queue is not received in the failover wait time limit. You can set this field to **Proceed with failover** or to **Cancel failover**.

8. On the **Exit Program** page, you can specify the information for a CRG exit program. Exit programs are required all CRG types except for device CRGs. Exit programs are called after a cluster-related event for a CRG occurs and responds to that event.
9. On the **Recovery Domain** page, add nodes to the recovery domain and specify their role within the cluster.

Related information:

Create Cluster Resource Group (CRTCRG) command

Create Cluster Resource Group (QcstCreateClusterResourceGroup) API

Creating data CRGs:

Data cluster resource groups (CRGs) are primarily used with logical replication applications, which are provided by several high availability business partners. If you are implementing a high-availability solution based on logical replication you can create a data CRG to assist the replication of data between primary and backup nodes.

To create a data CRG, complete the following steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, select **Work with Cluster Resource Groups** to show a list of cluster resource groups in the cluster.
5. On the Cluster Resource Group page, click the **Select Action** menu.
6. Select **New Data CRG** and click **Go**. The New Data CRG page displays.
7. On the **General** page, specify the following information regarding the data CRG:
   - In the **Name** field, specify the name of the CRG. The name cannot exceed 10 characters.
   - In the **Description** field, enter a description of the CRG. The description cannot exceed 50 characters.
   - In the **Distributed information user queue** field indicate the name of the user queue to receive distributed information. The name cannot exceed 10 characters. In the **Library** field specify the name of the library that contains the user queue to receive the distributed information. The library name cannot be *CURLIB, QTEMP, or *LIBL. The name cannot exceed 10 characters.

     Note: If you set the Distribute information user queue to blank, you must also set the Library name to blank, the Failover wait time to 0, and the Failover default action to 0.
   - In the **Failover message queue** field, specify the name of the message queue to receive messages when a failover occurs for this cluster resource group. If this field is set, the specified message queue must exist on all nodes in the recovery domain after the exit program is completed. The failover message queue cannot be in an independent disk pool. In the **Library** field, specify the name of the library that contains the message queue to receive the failover message. The library name cannot be *CURLIB, QTEMP, or *LIBL.
   - In the **Failover wait time** field, specify the number of minutes to wait for a reply to the failover message on the cluster message queue. Possible values include:

     Do not wait
     Failover proceeds without user intervention.

     Wait forever
     Failover waits forever until a response is received to the failover inquiry message.

     number
     Specify the number of minutes to wait for a response to the failover inquiry message. If a response is not received in the specified number of minutes, the value in the Failover default action field specifies how to proceed.
8. On the **Exit Program** page, you can specify the information for a CRG exit program. Exit programs are required all CRG types except for device CRGs. Exit programs are called after a cluster-related event for a CRG occurs and responds to that event.
9. On the **Recovery Domain** page, add nodes to the recovery domain and specify their role within the cluster.

Related information:

Create Cluster Resource Group (CRTCRG) command

Create Cluster Resource Group (QcstCreateClusterResourceGroup) API

Creating device CRGs:

A device cluster resource group (CRG) is made up of a pool of hardware resources that can be switched as an entity. To create switchable devices within a high-availability solution, the nodes that use these devices need to be a part of a device CRG.

Prior to creating a device CRG, add all nodes that will share a switchable resource to a device domain.

To create a device CRG, complete the following steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, select **Work with Cluster Resource Groups** to show a list of cluster resource groups in the cluster.
5. On the Cluster Resource Group page, click the **Select Action** menu.
6. Select **New Device CRG** and click **Go**. The **New Device CRG** wizard is shown. The **New Device CRG** task is only available if all the nodes in the recovery domain are started.
7. Follow the instructions in the **New Device CRG** wizard to create the new device CRG. While running this wizard, you can create a new device CRG. You can also create either a new independent disk pool or specify an existing disk pool to use.

   The device CRG keeps the hardware resource information identical on all recovery domain nodes and verifies that the resource names are identical. You can also configure a cluster administrative domain to keep the enrolled attributes of the configuration objects, which might include resource names, identical across the cluster administrative domain. If you are using cross-site mirroring, you should create separate device CRGs for independent disk pools and other types of switchable devices at each site.

Related information:

Create Cluster Resource Group (CRTCRG) command

Create Cluster Resource Group (QcstCreateClusterResourceGroup) API

Creating peer CRGs:

You can create a peer CRG to define node roles in load-balancing environments.

To create a peer CRG in a cluster, complete the following steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, select **Work with Cluster Resource Groups** to show a list of cluster resource groups in the cluster.
5. On the Cluster Resource Group page, click the **Select Action** menu.
6. Select **New Peer CRG** and click **Go**. The New Peer CRG page is shown.
7. On the **General** page, specify the following information regarding the peer CRG:
   - In the **Name** field, specify the name of the CRG. The name cannot exceed 10 characters.
   - In the **Description** field, enter a description of the CRG. The description cannot exceed 50 characters.
   - In the **Application ID** field, specify the application identifier for the peer cluster resource groups in the format *[VendorName].[ApplicationName]*. For example, MyCompany.MyApplication. The identifier cannot exceed 50 characters.

8. On the **Exit Program** page, you can specify the information for a CRG exit program. Exit programs are required all CRG types except for device CRGs. Exit programs are called after a cluster-related event for a CRG occurs and responds to that event.
9. On the **Recovery Domain** page, add nodes to the recovery domain and specify their role within the cluster.

Related information:

Create Cluster Resource Group (CRTCRG) command

Create Cluster Resource Group (QcstCreateClusterResourceGroup) API

Configuring cluster administrative domains

In a high-availability environment, it is necessary that the application and operational environment remain consistent among the nodes that participate in high availability. Cluster administrative domain is the i5/OS implementation of environment resiliency and ensures that the operational environment remains the consistent across nodes.

Creating a cluster administrative domain

In a high-availability solution, the cluster administrative domain provides the mechanism that keeps resources synchronized across systems and partitions within a cluster.

To create the cluster administrative domain, a user must have *IOSYSCFG authority and authority to the QCLUSTER user profile. To manage a cluster administrative domain, a user must be authorized to the CRG that represents the cluster administrative domain, the QCLUSTER user profile, and cluster resource group commands.

To create a cluster administrative domain, complete these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, click **Work with Administrative Domains** to list cluster administrative domains in the cluster. If no cluster administrative domains have been configured, this list is empty.
5. On the **Administrative Domain** tab, select **New Administrative Domain**.
6. In the New Administrative Domain page, specify the following information on the cluster administrative domain:
   - In the **Name** field, enter the name of the cluster administrative domain. The name cannot exceed 10 characters.
   - The **Cluster** field displays the name of the cluster. You cannot change the value of this field.
   - In the **Synchronization option** field, specify the synchronization behavior when a node joins a cluster administrative domain. This field is enabled only if the cluster is at version 6 or greater. Possible values follow:

     Last Change Option (default)
     Select this option if all changes to monitored resources should be applied to a cluster administrative domain. The most recent change that is made to a monitored resource is applied to the resource on all active nodes.

     Active Domain Option
     Select this option if only changes to monitored resources are allowed from active nodes. Changes made to monitored resources on inactive nodes are discarded when the node joins the cluster administrative domain. The Active Domain option does not apply to network server storage spaces (*NWSSTG) or network server configurations (*NWSCFG). Synchronization of these resources is always based on the last change that was made.

- From the **Nodes in the administrative domain** list, select the nodes that you would like to add to the cluster administrative domain and select **Add**.

Related concepts:

"Maintaining user profiles on all nodes" on page 79
You can use two mechanisms for maintaining user profiles on all nodes within a cluster.

Related information:

Create Cluster Administrative Domain (CRTCAD) command

Create Cluster Administrative Domain (QcstCrtClusterAdminDomain) API

Adding a node to the cluster administrative domain

You can add additional nodes to a cluster administrative domain within a high-availability solution.

Before adding a node to a cluster administrative domain, ensure that node is also part of the cluster in which the cluster administrative domain resides. If it is not, you cannot add the node to the cluster administrative domain. The cluster administrative domain does not have to be active, but the resources will just not be made consistent until it is active.

When you add a node to the administrative domain, the MREs from the domain are copied to the node being added. If the monitored resource does not exist on the new node, it is created by the cluster administrative domain. If the monitored resource already exists on the node being added, it is synchronized with the rest of the cluster administrative domain if the domain is active. That is, the values of the attributes for each monitored resource on the node that is joining are changed to match the global values for the monitored resources in the active domain.

To add a node to a cluster administrative domain, complete these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, click **Work with Administrative Domains** to display a list of cluster administrative domains in the cluster.
5. On the Administrative Domains page, select a cluster administrative domain.
6. From the **Select Action** menu, select **Properties**.
7. On the **Properties** page, choose the node that you want to add to the cluster administrative domain from the list of **Nodes in the administrative domain**. Click **Add**.

Related information:

Add Cluster Administrative Domain Node Entry (ADDCADNODE) command

Add Node To Recovery Domain (QcstAddNodeToRcvyDomain) API

Starting a cluster administrative domain

Cluster administrative domains provide environment resiliency for resources within an i5/OS high-availability solution.

When the cluster administrative domain is started, any change made to any monitored resource while the cluster administrative domain was ending is propagated to all active nodes in the cluster administrative domain.

To start a cluster administrative domain, complete the following steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.

4. On the Cluster Resource Services page, click **Work with Administrative Domains** to display a list of cluster administrative domains in the cluster.
5. On the Administrative Domains page, select a cluster administrative domain.
6. From the **Select Action** menu, select **Start**.

The Status column shows that the cluster administrative domain is started.

Related concepts:

"Synchronization of monitored resource"
Synchronization of monitored resources occurs when monitored resources are changed on nodes which have been defined in the cluster administrative domain.

Related information:

Start Cluster Administrative Domain (STRCAD) command

Synchronization of monitored resource

Synchronization of monitored resources occurs when monitored resources are changed on nodes which have been defined in the cluster administrative domain.

During this synchronization process, the cluster administrative domain attempts to change each resource with attributes whose values do not match its global values, unless there is a pending change for that resource. Any pending change is distributed to all active nodes in the domain and applied to each affected resource on each node. When the pending changes are distributed, the global value is changed and the global status of each affected resource is changed to *consistent* or *inconsistent*, depending on the outcome of the change operation for the resource on each node. If the affected resource is changed successfully on every active node in the domain, the global status for that resource is *consistent*. If the change operation failed on any node, the global status is set to *inconsistent*.

If changes are made to the same resource from multiple nodes while the cluster administrative domain is inactive, all of the changes are propagated to all of the active nodes as part of the synchronization process when the domain is started. Although all pending changes are processed during the activation of the cluster administrative domain, there is no guaranteed order in which the changes are processed. If you make changes to a single resource from multiple cluster nodes while the cluster administrative domains inactive, there is no guaranteed order to the processing of the changes during activation.

If a node joins an inactive cluster administrative domain (that is, the node is started while the cluster administrative domain is ended), the monitored resources are not resynchronized until the cluster administrative domain is started.

Note: The cluster administrative domain and its associated exit program are IBM-supplied objects. They should not be changed with the QcstChangeClusterResourceGroup API or the Change Cluster Resource Group (CHGCRG) command, or unpredictable results will occur.

After a cluster node that is part of a cluster administrative domain is ended, monitored resources can still be changed on the inactive node. When the node is started again, the changes will be resynchronized with the rest of the cluster administrative domain. During the resynchronization process, the cluster administrative domain applies any changes from the node that was inactive to the rest of the active nodes in the domain, unless changes had also been made in the active domain while the node was inactive. If changes were made to a monitored resource both in the active domain and on the inactive node, the changes made in the active domain are applied to the joining node. In other words, no changes made to any monitored resource are lost, regardless of the status of the node. You can specify the synchronization option to control synchronization behavior.

If you want to end a cluster node that is part of a cluster administrative domain, and not allow changes made on the inactive node to be propagated back to the active domain when the node is started (for example, when ending the cluster node to do testing on it), you must remove the node from the administrative domain peer CRG before you end the cluster node.

Related concepts:

Remove Admin Domain Node Entry (RMVCADNODE) command

Related tasks:

"Starting a cluster administrative domain" on page 111
Cluster administrative domains provide environment resiliency for resources within an i5/OS high-availability solution.

Related information:

Remove CRG Node Entry (RMVCRGNODE) command

Adding monitored resource entries

You can add a monitored resource entry (MRE) to a cluster administrative domain. Monitored resource entries define critical resources so that changes made to these resources are kept consistent across a high-availability environment.

To add a monitored resource entry, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. On the Cluster Resource Services page, click **Work with Administrative Domains** to display a list of cluster administrative domains in the cluster.
4. On the Administrative Domains page, click the context icon next to the cluster administrative domain name, and select **Monitored Resource Entries**.

   Note: The **Monitored Resource Entries** action is available only if the node that you are managing is part of the cluster administrative domain. The current list of monitored resource types is shown.
5. In the list of monitored resource types, click the context icon next to the monitored resource type, and select **Add Monitored Resource Entry**. The Add Monitored Resource Entry page is shown.
6. Select the attributes to be monitored for the monitored resource entry, and click **OK**. If the MRE object is in a library, you must specify the name and library for the object. The new monitored resource entry is added to the list of resources that the cluster administrative domain is monitoring. Changes to the monitored resource are synchronized across all active nodes in the cluster administrative domain when the domain is active. By default, all attributes associated with a monitored resource type are monitored; however, you can control what attributes are monitored by selecting attributes to be monitored.

Related tasks:

"Selecting attributes to monitor" on page 152
After you have added monitored resource entries, you can select attributes associated with that resource to be monitored by the cluster administrative domain.

Related information:

Add Admin Domain MRE (ADDCADMRE) command

Add Monitored Resource Entry (QfpadAddMonitoredResourceEntry) API

Adding monitored resource entries

You can add a monitored resource entry (MRE) to a cluster administrative domain. Monitored resource entries define critical resources so that changes made to these resources are kept consistent across a high-availability environment.

To add a monitored resource entry, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. On the Cluster Resource Services page, click **Work with Administrative Domains** to display a list of cluster administrative domains in the cluster.

4. On the Administrative Domains page, click the context icon next to the cluster administrative domain name, and select **Monitored Resource Entries**.

   Note: The **Monitored Resource Entries** action is available only if the node that you are managing is part of the cluster administrative domain. The current list of monitored resource types is shown.
5. In the list of monitored resource types, click the context icon next to the monitored resource type, and select **Add Monitored Resource Entry**. The Add Monitored Resource Entry page is shown.
6. Select the attributes to be monitored for the monitored resource entry, and click **OK**. If the MRE object is in a library, you must specify the name and library for the object. The new monitored resource entry is added to the list of resources that the cluster administrative domain is monitoring. Changes to the monitored resource are synchronized across all active nodes in the cluster administrative domain when the domain is active. By default, all attributes associated with a monitored resource type are monitored; however, you can control what attributes are monitored by selecting attributes to be monitored.

Related tasks:

"Selecting attributes to monitor" on page 152
After you have added monitored resource entries, you can select attributes associated with that resource to be monitored by the cluster administrative domain.

Related information:

Add Admin Domain MRE (ADDCADMRE) command

Add Monitored Resource Entry (QfpadAddMonitoredResourceEntry) API

Configuring switched disks

Switched disks are independent disk pools which have been configured as part of an i5/OS cluster. Switched disks allow data and applications stored within an independent disk pool to be switched to another system.

Creating an independent disk pool

To create an independent disk pool, you can use the New Disk Pool wizard. This wizard can assist you in creating a new disk pool and adding disk units to it.

With the New Disk Pool wizard you can include unconfigured disk units in a parity set, and you can start device parity protection and disk compression. As you add disk units, do not spread disk units that are in same parity set across multiple disk pools, because failure to one parity set would affect multiple disk pools.

Use the New Disk Pool wizard to create an independent disk pool using IBM Systems Director Navigator for i5/OS, follow these steps:

Note: To work with disk within the IBM Systems Director Navigator for i5/OS, you must have the appropriate password configuration for Dedicated Service Tools.

IBM Systems Director Navigator for i5/OS

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
4. Select **Disk Units**.
5. From the **Select Actions** menu, select **New Disk Pool**.
6. Follow the wizard's instructions to add disk units to a new disk pool.
7. Print your disk configuration to have it available in a recovery situation.
8. Record the relationship between the independent disk pool name and number.

System i Navigator

To use the New Disk Pool wizard to create an independent disk pool using System i Navigator, follow these steps:

1. In System i Navigator, expand **My Connections** (or your active environment).
2. Expand the system you want to examine, and expand **Configuration and Service** > **Hardware** > **Disk Units**.
3. Right-click **Disk Pools** and select **New Disk Pool**.
4. Follow the wizard's instructions to add disk units to a new disk pool.
5. Print your disk configuration to have it available in a recovery situation.
6. Record the relationship between the independent disk pool name and number.

Note: Add independent disk pools when your system is fully restarted. If you must use the New Disk Pool wizard in the dedicated service tools (DST) mode, you need to create an associated device description for the independent disk pool when the system is fully restarted. Use the Create Device Description (ASP) (CRTDEVASP) command to create the device description. Name the device description and resource name the same as you name the independent disk pool. You can use the Work with Device Descriptions (WRKDEVD) command to verify that the device description and independent disk pool name match.

Starting mirrored protection

The Add Disk Unit and New Disk Pool wizards guide you through the process of adding pairs of similar capacity disk units to a protected disk pool. When you have your disks configured correctly, you are ready to start mirroring for mirrored protection.

Mirrored protection is local to a single system and is distinct from cross-site mirroring. If you want to start mirroring on an independent disk pool that is unavailable, you can do so when your system is fully restarted. For all other disk pools, you need to restart your system to the dedicated service tools (DST) mode before starting mirrored protection.

There are restrictions to follow when starting mirror-protection on the load source disk unit.

- The smaller capacity disk must start out as the load source device when two disks of unequal capacity are matched as a mirrored pair,. The load source can then be matched with the larger capacity disk unit. For example, if the load source disk unit is a 35 G-Byte disk, it can be matched with a 36 GB disk. If the load source is a 36 G-Byte disk, it cannot be matched with the 35 G-Byte disk.
- The system must be directed to match the load source disk unit with a disk unit that is in a physical location the service processor cannot use to IPL the partition. From SST, select **Work with disk units->Work with disk configuration->Enable remote load source mirroring**. The **Enable remote load source mirroring** function allows a disk unit to be matched with the load source disk unit even though the disk unit resides in a physical location the service processor is not able to use to IPL the partiton.

To start mirroring using IBM Navigator for i, follow these steps:

1. Select **Configuration and Service** from your IBM Navigator for i window.
2. Select **Disk Pools**.
3. Select the disk pool that you want to mirror.
4. From the **Select Actions** menu, select **Start Mirroring**.

To start mirroring using System i Navigator, follow these steps:

1. In System i Navigator, expand **My Connections** (or your active environment).
2. Expand the System i you want to examine,**Configuration and Service** > **Hardware** > **Disk Units** > **Disk Pools**.
3. Right-click the disk pools you want to mirror, and select **Start Mirroring**.

Stopping mirrored protection

When you stop mirrored protection, one disk unit from each mirrored pair is unconfigured. Before you can stop mirrored protection for a disk pool, at least one disk unit in each mirrored pair in that disk pool must be present and active.

To control which mirrored disk unit of each pair is unconfigured, you may suspend the disk units that you want to become unconfigured. For disk units that are not suspended, the selection is automatic.

If you want to stop mirroring on an independent disk pool that is unavailable, you can do so when your system is fully restarted. For all other disk pools, you need to restart your system to the dedicated service tools (DST) mode before stopping mirrored protection.

Mirrored protection is dedicated to a single system, and is distinct from cross-site mirroring.

To stop mirrored protection using IBM Systems Director Navigator for i5/OS, follow these steps:

1. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
2. Select **Disk Pools**.
3. Select the disk pool that you want to stop.
4. From the **Select Actions** menu, select **Stop Mirroring**.

To stop mirrored protection using System i Navigator, follow these steps:

1. In System i Navigator, expand **My Connections** (or your active environment).
2. Expand the System i you want to examine,**Configuration and Service** > **Hardware** > **Disk Units** > **Disk Pools**.
3. Select the disk unit for which you want to stop mirrored protection.
4. Right-click any selected disk pool, and select **Stop Mirroring**.
5. Click **Stop Mirroring** from the resulting confirmation dialog box.

Adding a disk unit or disk pool

The Add Disk Unit wizard allows you to use an existing disk pool to add new or non-configured disk units.

The Add Disk Unit and Disk Pool wizards save you time by bundling several time-consuming configuration functions into one efficient process. They also take the guesswork out of disk unit configuration because they understand the capabilities of your system and only offer valid choices. For instance, the wizard does not list the option to start compression unless your system has that capability.

When you choose to add disk units to a protected disk pool, the wizard forces you to include the disk units in device parity protection or to add enough disk units of the same capacity to start mirrored protection. The wizard also gives you the option of balancing data across the disk pool or starting disk compression if these are permissible actions for your system configuration. You decide which options to choose so that the operation is tailored to your system.

To add a disk unit or disk pool using IBM Systems Director Navigator for i5/OS, follow these steps:

1. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
2. Select **Disk Units**.
3. From the **Select Actions** menu, select **Add Disk Unit**.
4. Follow the wizard's instructions to add disk units to your disk pool.

To add a disk unit or disk pool using System i Navigator, follow these steps:

1. In System i Navigator, expand **My Connections** (or your active environment).

2. Expand the System i you want to examine,**Configuration and Service** > **Hardware** > **Disk Units**.
3. To add disk units, right-click **All Disk Units** and select **Add Disk Unit**.
4. Follow the instructions in the wizard to complete the task.

Evaluating the current configuration

Before you change the disk configuration of your system, it is important to know exactly where the existing disk units are located in relation to disk pools, IOAs, and frames.

The graphical view of System i Navigator eliminates the process of compiling all this information by providing a graphical representation of how your system is configured. You can use the graphical view to perform any function that is possible through the Disk Units list view of System i Navigator, with the added benefit of being able to see a visual representation. If you right-click any object in the table, such as a specific disk unit, disk pool, parity set, or frame, you see the same options as in the main System i Navigator window.

You can choose how to view the hardware in the Disk Unit Graphical View window. For example, you can select to view by disk pools, and then select a disk pool in the list to display only those frames that contain the disk units that make up the selected disk pool. You can select Show all frames to see whether or not they contain disk units in the selected disk pool. You can also select Show device positions to associate disk unit names with the device position where they are inserted.

You can right-click any highlighted blue disk unit in the graphical view and select an action to perform on the disk unit. For example, you can select to start or stop compression on a disk unit, include or exclude the disk unit in a parity set, or rename the disk unit. If the disk unit has mirrored protection, you can suspend or resume mirroring on the disk unit. If you right-click an empty disk unit slot, you can start the Install Disk Unit wizard.

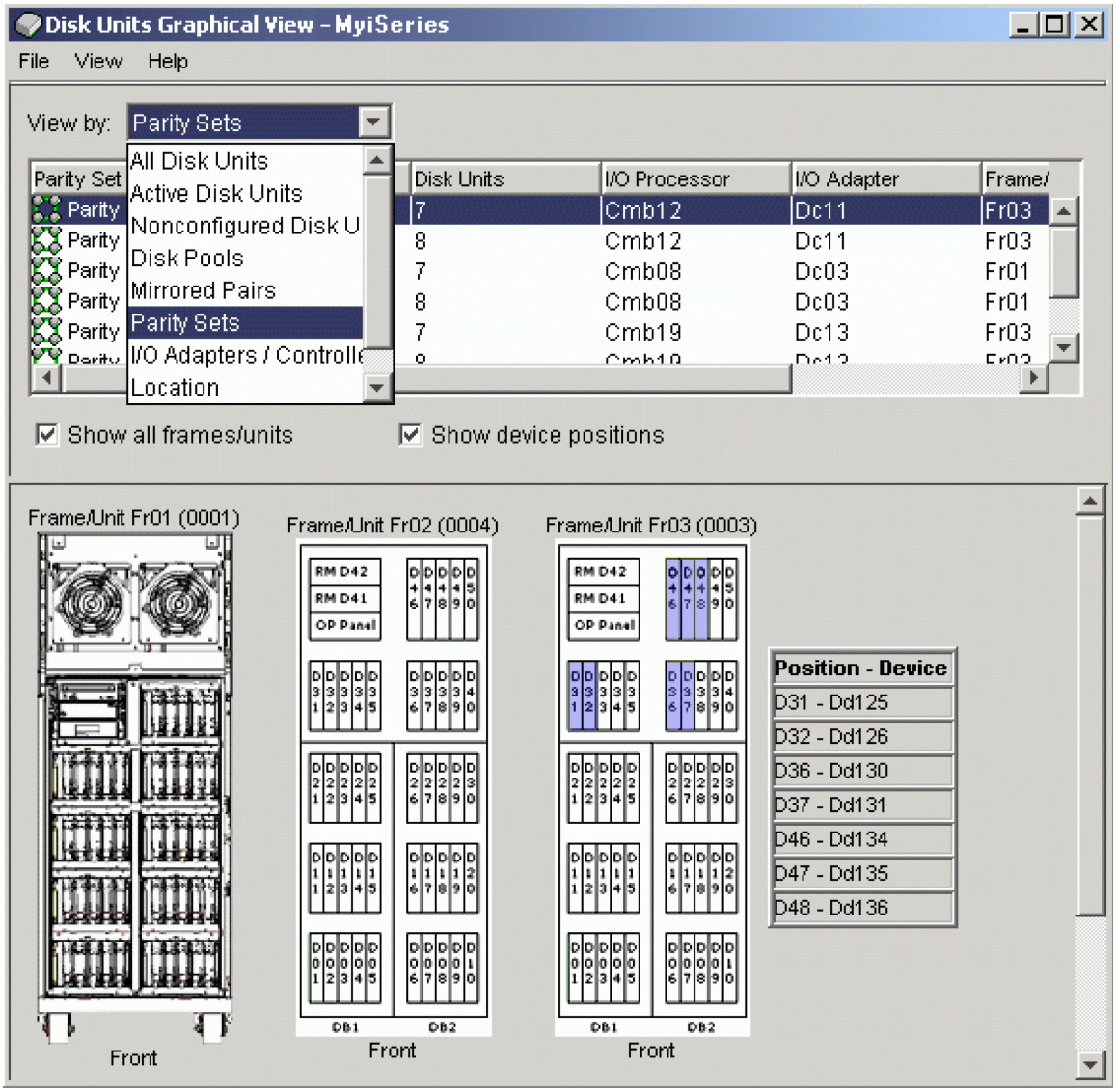
To activate the graphical view from System i Navigator, follow these steps:

1. In System i Navigator, expand **My Connections** (or your active environment).
2. Expand the system you want to examine, **Configuration and Service** > **Hardware** > **Disk Units**
3. Right-click **All Disk Units**, and select **Graphical View**

To activate the graphical view from IBM Navigator for i, follow these steps:

1. Select **Configuration and Service** from the IBM Navigator for i window.
2. Select **Disk Units** or **Disk Pools**.
3. From the **Select Actions** menu, select **Graphical View**.

Here is an example of the graphical view in System i Navigator. The View by menu lists several options for viewing disk units.



Making a disk pool available

To access the disk units in an independent disk pool, you must make the disk pool available (vary it on).

To access the disk units in an independent disk pool and the objects in the corresponding database, you must make the disk pool available (vary it on). If you are using geographic mirroring, you must make the production copy of the disk pool available. You can only make the mirror copy available if it is detached. For a geographically mirrored disk pool, you must also make sure that the switchable hardware group is started before attempting to make the disk pool available unless geographic mirroring is suspended.

In a multisystem clustered environment, you can make the disk pool available to the current node or to another node in the cluster. The independent disk pool can only be varied on for one node at a time. When you want to access the independent disk pool from a different node, you must switch the independent disk pool to the backup cluster node. See Performing a switchover for details on switching a device CRG (referred to as a switchable hardware group in System i Navigator) to the backup node.

Note: If you make a primary or secondary disk pool available, all of the disk pools in the disk pool group are also made available at the same time.

When you make a disk pool available or perform disk configuration changes on an independent disk pool, processing can seem to stop. If you are doing other device description activities, then make available and disk configuration changes will wait.

Failures early in make available processing of a geographically mirrored disk pool might cause a full synchronization on the next make available or resume.

To make an independent disk pool available:

1. In System i Navigator, expand **My Connections** (or your active environment).
2. Expand the you want to examine,**Configuration and Service** > **Hardware** > **Disk Units**.
3. Expand **Disk Pools**.
4. Right-click the unavailable disk pool and select **Make Available**. You can select multiple disk pools to make available at the same time.
5. From the dialog box displayed, click **Make Available** to make the disk pool available.

You can use the Vary Configuration (VRYCFG) command in the character-based interface to make the disk pool available.

Use the Display ASP Status (DSPASPSTS) command to identify where a step is in the process.

Configuring cross-site mirroring

Cross-site mirroring is a collective term used for several different high availability technologies, including geographic mirroring, metro mirror and global mirror. Each one of these technologies has specific tasks related to configuration.

Configuring geographic mirroring

Geographic mirroring is a sub-function of cross-site mirroring. To configure a high-availability solution by using geographic mirroring, you need to configure a mirroring session between the production system and the backup system.

Before configuring geographic mirroring, you must have an active cluster, nodes, and CRG. The independent disk pools which you plan to use for geographic mirroring must also be varied off (unavailable) to complete configuration. The topic, Scenario: Cross-site mirroring with geographic mirroring, provides step-by-step instructions for setting up a high-availability solution based on geographic mirroring.

IBM Navigator for i

To configure geographic mirroring by using IBM Systems Director Navigator for i, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i window.
4. Select **Disk Pools**.
5. Select the disk pool that you want to use as the production (source) copy.
6. From the **Select Actions** menu, select **New Session**.
7. Follow the wizard's instructions to complete the task.

System i Navigator

To configure geographic mirroring by using System i Navigator, follow these steps:

1. In System i Navigator, expand **My Connections** (or your active environment).

2. Expand the system that you want to use as the production copy.
3. Expand **Configuration and Service** > **Hardware** > **Disk Units** > **Disk Pools**.
4. Right-click the disk pool that you want to use as the production copy and select **Sessions** > **New**.
5. Follow the wizard's instructions to complete the task.

Related concepts:

"Scenario: Switched disk with geographic mirroring" on page 83
This scenario describes an i5/OS high-availability solution that uses switched disks with geographic mirroring in a three-node cluster. This solution provides both disaster recovery and high availability.

Configuring metro mirror session

For i5/OS high availability solutions that use IBM System Storage metro mirror technology, you need to configure a session between the System i machine and IBM System Storage external storage units that have metro mirror configured. In i5/OS, metro mirror sessions do not set up the mirroring on the external storage units, but rather sets up a relationship between the i5/OS systems and the existing metro mirror configuration on external storage units.

Before creating a metro mirror session on i5/OS, you should have configured metro mirror on the IBM System Storage external storage units. For information about using metro mirror on IBM System Storage DS8000, see IBM System Storage DS8000 Information Center.

To configure metro mirror session, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
4. Select **Disk Pools**.
5. Select the disk pool that you want to use as the production (source) copy.
6. From the **Select Actions** menu, select **New Session**.
7. Follow the wizard's instructions to complete the task.

Related information:

Add ASP Copy Description (ADDASPCPYD) command

Start ASP Session (STRASPSSN) command

Configuring global mirror session

For i5/OS high availability solutions that use IBM System Storage global mirror technology, you need to configure a session between the System i machine and IBM System Storage external storage units that have global mirror configured. In i5/OS, global mirror sessions do not set up the mirroring on the external storage units, but rather sets up a relationship between the i5/OS systems and the existing global mirror configuration on external storage units.

IBM System Storage global mirror technology requires all users to share one global mirror connection. i5/OS high availability global mirror allows only one System i partition to configure global mirror on a given System Storage server. No other System i partitions or servers from other platforms may use global mirror at the same time. Adding more than one user to a global mirror session will cause unpredictable results to occur.

Before creating a global mirror session on i5/OS, you should have configured global mirror on the IBM System Storage external storage units. For information about using global mirror on IBM System Storage DS8000, see IBM System Storage DS8000 Information Center.

To configure global mirroring, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
4. Select **Disk Pools**.
5. Select the disk pool that you want to use as the production (source) copy.
6. From the **Select Actions** menu, select **New Session**.
7. Follow the wizard's instructions to complete the task.

Related information:

Add ASP Copy Description (ADDASPCPYD) command

Start ASP Session (STRASPSSN) command

Chapter 5. Managing high availability

After you have configure an i5/OS high-availability solution, you can manage that solution by using several interfaces that are related to high availability.

Scenarios: Managing high availability solutions

As a system operator or administrator of your high-availability solution, you need to perform common tasks like backup and system maintenance in your high-availability environment.

The following scenarios provide instructions on performing common system tasks, such as backups and upgrades, as well as examples of managing high-availability events, such as cluster partitions and failover. For each scenario, a model environment has been chosen. The instructions for each scenario correspond to that particular high-availability solution and are meant for example purposes only.

Scenarios: Performing backups in a high-availability environment

Depending on your high availability-solution and your backup strategy, the method for backing up data can be different. However, there is a common set of tasks when you perform backup operations for systems in a high availability environment.

In several high availability-solutions, you have the capability of performing remote backups from the second copy of data that is stored on the backup system. Remote backups allow you to keep your production system operational, while the second system is backed up. Each of these scenarios provides examples of two high-availability solutions where backups are performed remotely on the backup system.

In the first scenario, remote backups are performed in a high availability solution that uses geographic mirroring technology. The second scenario shows how the FlashCopy feature can be used in a high-availability environment that use IBM System Storage solutions, such as metro or global mirror.

Scenario: Performing backups in geographic mirroring environment

This scenario provides an overview of tasks that are necessary when performing a remote backup in a i5/OS high-availability solution that uses geographic mirroring.

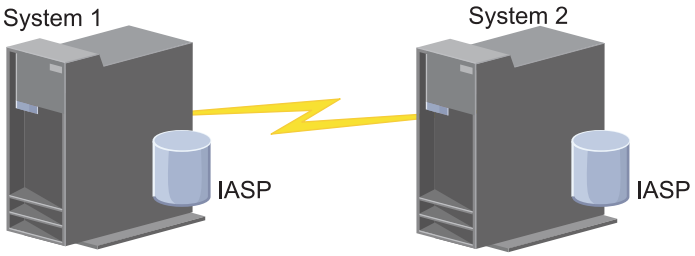
Overview

In this example, a system administrator needs to perform a backup of data stored on independent disk pools that are used in a high-availability solution based on geographic mirroring technology. The administrator does not want to affect the production system by taking it offline to perform the backup. Instead the administrator plans to temporarily detach the mirrored copy, and then perform a backup from the second copy of data located on independent disk pools at a remote location.

Note: Detaching the mirrored copy essentially ends geographic mirroring until the copy is reattached to the production. During the time it is detached, high availability and disaster recovery are not operational. If an outage to the production system occurs during this process, some data will be lost.

Details

The following graphic illustrates this environment:



Configuration steps

1. Quiescing independent disk pool
2. "Detaching mirror copy" on page 172
3. Making disk pool available
4. Backing up independent disk pool
5. "Resuming an independent disk pool" on page 170
6. "Reattaching mirror copy" on page 173

Scenario: Performing a FlashCopy function

In this example an administrator wants to perform a backup from the remote copy of data stored in an external storage units at the backup site. Using the FlashCopy function available with IBM Storage Solutions, the administrator reduces his backup time considerably.

Overview

In this example, a system administrator needs to perform a backup of data stored on IBM System Storage external storage units. The administrator does not want to affect the production system by taking it offline to perform the backup. Instead the administrator plans to perform a FlashCopy operation, which takes a point-in-time capture of the data. From this data, the administrator backs up the data to external media. The FlashCopy operation only takes a few seconds to complete, thus reducing the time for the entire backup process.

Although in this example the FlashCopy feature is being used to for backup operations, it should be noted that the FlashCopy feature has multiple uses. For example, FlashCopy can be used for data warehousing to reduce query workload on production systems, or for duplicating production data to create a test environment.

Configuration steps

1. "Quiescing an independent disk pool" on page 169
2. "Configuring a FlashCopy session" on page 181
3. Perform the FlashCopy function on IBM System Storage external storage units. For information about using the FlashCopy function on IBM System Storage DS8000, see IBM System Storage DS8000 Information Center .
4. "Resuming an independent disk pool" on page 170
5. Make disk pool available
6. Backing up independent disk pool

Scenario: Upgrading operating system in a high-availability environment

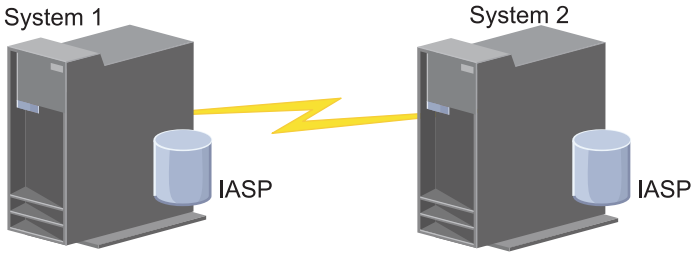
In this example, a system administrator is upgrading the operating system for two IBM i systems in a high-availability solution based on geographic mirroring.

Overview

The system administrator needs to upgrade the operating system for two systems in the high-availability environment. In this example, there are two nodes: System 1 and System 2. System 1 is the production copy and System 2 is the mirror copy. Both systems are at IBM i V6R1. The independent disk pool is online, geographic mirroring is active, and the systems are synchronized. The system administrator wants to upgrade both of these systems to i 7.1.

Details

The following graphic illustrates the environment:



Configuration steps

1. Detach the mirror copy (System 2).
2. End the CRG (System 2).
3. Stop the node (System 2).
4. Upgrading System 2 to the new release. See Upgrading or replacing i5/OS and related software for details.
5. Install IBM PowerHA for i licensed program.
6. Make disk pool available and test applications on System 2. Testing the applications ensures that they operate as expected within the new release. After application tests are completed, you can finish the upgrade by completing the rest of these steps.
7. Make disk pool unavailable on the detached mirrored copy (System 2).
8. Reattach mirrored copy. This initiates a resynchronization of the mirrored data. After the resynchronization is completed, you can continue the upgrade process.
9. Performing switchovers. This makes the mirrored copy (System 2) the new production copy and the production copy (System 1) becomes the new mirrored copy.

   Note: Geographic mirroring is suspended because you cannot perform geographic mirroring from n-1 to n. You can perform geographic mirroring from n to n-1 without problems. In this scenario, geographic mirroring is suspended after a switchover is completed. Data is now no longer mirrored during the remainder of the upgrade process because there is no longer a valid backup system.
10. End the CRG (System 1).
11. Stop the node (System 1).

12. Upgrade System 1 to the new release. See Upgrading or replacing i5/OS and related software for details.
13. Install IBM PowerHA for i licensed program.
14. Start nodes (System 1).
15. Start CRGs (System 1).
16. Resume mirroring
17. Perform switchover. This switches the current mirrored copy (System 1) back to the production copy and the production copy (System 2) to become the mirrored copy. This is the original configuration before the upgrade.
18. Adjusting the cluster version of a cluster
19. Adjusting the high availability version of PowerHA LP

Example: Upgrading operating system

In a high-availability environments, you must perform specific actions prior to performing operating system upgrades.

The following examples can help you determine what you need to do to perform an upgrade in your cluster environment. Before performing the upgrade or any actions you should first determine the current cluster version for your cluster.

Notes:

1. V6R1 represents the current release of the operating system.
2. 7.1 represents the new release of the operating system.
3. V5R4 represents the prior release of the operating system.

Example 1: The node to be upgraded is at IBM i V6R1. All other nodes in the cluster are at IBM i V6R1 or higher. The current cluster version is 6.

Action:

1. Upgrade node to IBM i 7.1.
2. Start the upgraded node.

Example 2: The node to be upgraded is at IBM i V6R1. All other nodes in the cluster are at IBM i V6R1 or higher. The current cluster version is 5.

Action:

1. Upgrade the node to IBM i 7.1.
2. Start the upgraded node.

Example 3: The node to be upgraded is IBM i V5R4. All other nodes in the cluster are at IBM i V5R4 or higher. The current cluster version is 5.

Action:

1. Upgrade the node to IBM i 7.1.
2. Start the upgraded node.

Example 4: The node to be upgraded is at IBM i V5R4. All other nodes in the cluster are at IBM i V5R3 or higher. The current cluster version is 4.

Actions:

1. Upgrade all nodes to V5R4.
2. Start all of the upgraded nodes.
3. Change the cluster version to 5.
4. Upgrade the node to 7.1.
5. Start the upgraded node.

Example 5: The node to be upgraded is at IBM i V5R3 or lower. All other nodes in the cluster are at IBM i V5R3 or lower. The current cluster version cluster version is less than or equal to 4.

Actions:

1. Upgrade all nodes to V5R4.
2. Start all of the upgraded nodes.
3. Change the cluster version to 5.
4. Upgrade the node to 7.1.
5. Start the upgraded node.

The following table provides actions you need to take when performing an upgrade in a cluster environment.

Table 8. Upgrading nodes to IBM i 7.1

Current release of node you are upgrading	Current cluster version	Actions
V6R1	5 or 6	1. Upgrade the node to IBM i 7.1. 2. Start the upgraded node.
V5R4	5	1. Upgrade the node to IBM i 7.1. 2. Start the upgraded node.
V5R4	4	1. Upgrade all nodes to V5R4. 2. Start all of the upgraded node. 3. Change the cluster version to 5. 4. Upgrade the node to 7.1. 5. Start the upgraded node.
V5R3 or lower	less than or equal to 4	1. Upgrade all nodes to V5R4. 2. Start all of the upgraded node. 3. Change the cluster version to 5. 4. Upgrade the node to IBM i 7.1. 5. Start the upgraded node.

Scenario: Making a device highly available

In addition to independent disk pools, you can also provide high availability for other supported devices. In this situation, the high availability administrator wants to provide high availability to Ethernet lines.

Overview

The system administrator wants to provide high availability for Ethernet lines used within the high-availability solution. The current configuration provides high availability for planned outages with two systems that uses switched disk technology. This solution also use cluster administrative domain to manage and synchronize changes to the operational environment of the high availability solution. This example assumes that all high availability configuration and Ethernet configuration has been completed successfully prior to finishing these steps. It is also assumed that the current state of the high availability is active and all monitored resources are consistent within the environment. This example provides steps on configuring high availability for a Ethernet line.

Configuration steps

1. “Creating switchable devices” on page 141
2. “Adding monitored resource entries” on page 113
3. “Selecting attributes to monitor” on page 152

Managing clusters

Using the Cluster Resource Services graphical interfaces, you can perform many tasks associated with the cluster technology that is the basis of your i5/OS high availability solution. These tasks help you manage and maintain your cluster.

Some of the changes that you can make to the cluster after you configure it include the following:

Cluster tasks

- Add a node to a cluster
- Remove nodes from a cluster
- Start a cluster node
- End a cluster node
- Adjust the cluster version of a cluster to the latest level
- Delete a cluster
- Change cluster node

Cluster resource group tasks

- Create new cluster resource groups
- Delete existing cluster resource groups
- Start a cluster resource group
- Add a node to a cluster resource group
- Remove a node from a cluster resource group
- End a cluster resource group
- Change the recovery domain for a cluster resource group
- Perform a switchover
- Add a node to a device domain
- Remove a node from a device domain

Cluster administrative domain tasks

- Create a cluster administrative domain
- Add monitored resources
- Delete cluster administrative domain

Adjusting the PowerHA version

The PowerHA version is the version at which the nodes in the cluster managed by the PowerHA product are actively communicating with each other.

The PowerHA version values determine which functions can be used by the PowerHA product. The PowerHA version may require a certain cluster version in order to operate. For example, PowerHA version 2.0 requires a current cluster version of 7.

The current PowerHA version is set when a cluster is created. If a cluster exists, the current PowerHA version will be set to the lowest supported version.

Like cluster version, PowerHA has a current and potential version level. The current PowerHA version is the version at which the nodes in the cluster which are known by the PowerHA product are actively communicating with each other. The potential PowerHA version is the highest PowerHA version the node can support. The PowerHA version cannot be changed until all PowerHA nodes are installed with a common potential PowerHA version. The potential PowerHA version can be between n and n+1. For

example, NODE1 has a potential PowerHA version of 2.0, NODE2 has a potential PowerHA version of 2.0, and NODE3 has a potential PowerHA version of 3.0. All three nodes can support version 2.0, so the current PowerHA version can be adjusted to 2.0.

Beginning with PowerHA version 2.0, if a node with an incompatible potential PowerHA version is added to the cluster, the node will successfully be added, but the node will be considered "unknown" to PowerHA. If a node is unknown to PowerHA, certain product functions cannot be performed on the node. A node is known to PowerHA if the node has the PowerHA product installed, and the potential PowerHA version is compatible with the current PowerHA version.

The current PowerHA version can be changed with the Change Cluster Version (CHGCLUVER) command.

The Change Cluster Version (CHGCLUVER) command can only be used to adjust to a higher cluster or PowerHA version. If you want to adjust the PowerHA version by two, the CHGCLUVER command must be run twice.

The current cluster version cannot be set higher than the lowest potential node version in the cluster. Likewise, the current PowerHA version cannot be set higher than the lowest potential PowerHA version of any node in the cluster. PowerHA potential version of any node in the cluster. To view the potential node and PowerHA versions, use the Display Cluster Information (DSPCLUINF) command.

Use the following instructions to verify and change the cluster version for a node.

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, select the **Display Cluster Properties** task.
5. On the Cluster Properties page, click the **General** tab.
6. Verify the cluster version setting or change the version to the correct setting.
7. Verify the PowerHA version setting or change the version to the correct setting.

Related concepts:

Cluster version

Related information:

Change Cluster Version (CHGCLUVER) command

Adjusting the cluster version of a cluster

The cluster version defines the level at which all the nodes in the cluster are actively communicating with each other.

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

To change the cluster version, all nodes in the cluster must be at the same potential version. The cluster version can then be changed to match the potential version. This allows the new function to be used. The version can only be increased by one. It cannot be decremented without deleting the cluster and recreating it at a lower version. The current cluster version is initially set by the first node defined in the cluster. Subsequent nodes added to the cluster must be equal to the current cluster version or the next level version; otherwise, they cannot be added to the cluster.

If you are upgrading a node to a new release, you must ensure that the node has the appropriate cluster version. Cluster only supports a one version difference. If all the nodes in the cluster are at the same release, you should upgrade to the new release, before changing the cluster version. This ensures that all

functions associated with the new release are available. See the topic, "Scenario: Upgrading operating system in a high-availability environment" on page 125 for detailed actions for upgrading to a new release.

Use the following instructions to verify and change the cluster version for a node.

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, select the **Display Cluster Properties** task.
5. On the Cluster Properties page, click the **General** tab.
6. Verify the cluster version setting or change the version to the correct setting.

Related concepts:

Cluster version

Related information:

Change Cluster Version (CHGCLUVER) command

Adjust Cluster Version (QcstAdjustClusterVersion) API

Deleting a cluster

When you delete a cluster, cluster resource services ends on all active cluster nodes and they will be removed from the cluster.

You must have at least one active node before you can delete a cluster. If you have switched disks or other switchable devices in your cluster, you must first remove each node from the device domain before you delete your cluster. Otherwise, you might not be able to add the disks back into another cluster.

To delete a cluster, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the **Cluster Resource Services** page, click **Delete Cluster**.
5. The **Delete Cluster** confirmation window displays. Select **Yes** to delete the cluster. After you delete the cluster, the **Cluster Resource Services** page changes to display the **New Cluster** task.

Related tasks:

"Removing a node from a device domain" on page 135
A *device domain* is a subset of nodes in a cluster that share device resources.

Related information:

Delete Cluster (DLTCLU) command

Delete Cluster (QcstDeleteCluster) API

Displaying cluster configuration

You can display a detailed report that provides information on the cluster configuration. The cluster configuration report provides detailed information about the cluster, node membership list, configuration and tuning parameters, and each cluster resource group in the cluster.

To display cluster configuration, complete the following steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.

4. On the **Cluster Resource Services** page, select the **Display Configuration Information** task. This displays the Cluster Configuration and Properties page. You can save this page as a file or print it.

Related information:

Display Cluster Information (DSPCLUINF) command

Saving and restoring cluster configuration

If you use clustering on your systems, it is still important that you create a backup and recovery strategy to protect your data.

If you are planning on using clustering as your backup strategy so that you have one system up and running while the other system is down when its being backed up, it is recommended that you have a minimum of three systems in the cluster. By having three systems in the cluster, you will always have one system to switch over to should a failure occur.

Saving and restoring cluster resource groups

You can save a cluster resource group whether the cluster is active or inactive. The following restrictions apply for restoring a cluster resource group:

- If the cluster is up and the cluster resource group is not known to that cluster, you cannot restore the cluster resource group.
- If the node is not configured for a cluster, you cannot restore a cluster resource group.

You can restore a cluster resource group if the cluster is active, the cluster resource group is not known to that cluster, the node is in the recovery domain of that cluster resource group, and the cluster name matches that in the cluster resource group. You can restore a cluster resource group if the cluster is configured but is not active on that node and if that node is in the recovery domain of that cluster resource group.

Preparing for a disaster

In the event of a disaster, you might need to reconfigure your cluster. In order to prepare for such a scenario, it is recommended that you save your cluster configuration information and keep a hardcopy printout of that information.

1. Use the **Save Configuration (SAVCFG)** command or the **Save System (SAVSYS)** command after making cluster configuration changes so that the restored internal cluster information is current and consistent with other nodes in the cluster. See Saving configuration information for details on performing a SAVCFG or SAVSYS operation.
2. Print a copy of the cluster configuration information every time you change it. You can use the **Display Cluster Information (DSPCLUINF)** command to print the cluster configuration. Keep a copy with your backup tapes. In the event of a disaster, you might need to reconfigure your entire cluster.

Related information:

Saving configuration information

Save Configuration (SAVCFG) command

Save System (SAVSYS) command

Display Cluster Information (DSPCLUINF) command

Monitoring cluster status

The Cluster Resource Services graphical interface monitors cluster status and displays a warning message when nodes participating in the high availability solution become inconsistent.

The Cluster Resource Services graphical interface displays warning message HAI0001W on the Nodes page if the cluster is inconsistent. An inconsistent message means that information that is retrieved from this node might not be consistent with other active nodes in the cluster. Nodes become inconsistent when they become inactive within the cluster.

To obtain consistent information, you can either access the cluster information from an active node in the cluster, or start this node and retry the request.

To monitor cluster status, complete these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Node page, HAI0001W is shown if the node is inconsistent: `The local cluster node is not active. Cluster information may not be accurate until the local node has been started.`

Related tasks:

"Starting nodes" on page 100
Starting a cluster node starts clustering and cluster resource services on a node in an IBM i high availability environment.

Related information:

Display Cluster Information (DSPCLUINF) command

Display Cluster Resource Group Information (DSPCRGINF) command

List Cluster Information (QcstListClusterInfo) API

List Device Domain Info (QcstListDeviceDomainInfo) API

Retrieve Cluster Resource Services Information (QcstRetrieveCRSInfo) API

Retrieve Cluster Information (QcstRetrieveClusterInfo) API

List Cluster Resource Groups (QcstListClusterResourceGroups) API

List Cluster Resource Group Information (QcstListClusterResourceGroupInf) API

Specifying message queues

You can either specify a cluster message queue or a failover message queue. These message queues help you determine causes of failures in your i5/OS high availability environment.

A cluster message queue is used for cluster-level messages and provides one message which controls all cluster resource groups (CRGs) failing over to a specific node. A failover message queue is used for CRG-level messages and provides one message for each CRG that is failing over.

Specifying a cluster message queue

Note: You can also configure a cluster to use a cluster message queue by specifying the message queue while running the Create Cluster wizard.

To specify a cluster message queue, complete these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, click **Display Cluster Properties**.
5. On the Cluster Properties page, click **Cluster Message Queue**.
6. Specify the following information to create a cluster message queue:
   - In the **Name** field, specify the name of the message queue to receive messages that deal with a failover at a cluster or node level. For node-level failovers, one message is sent that controls the

failover of all cluster resource groups with the same new primary node. If a cluster resource group is failing over individually, one message is sent that controls the failover of that cluster resource group. The message is sent on the new primary node. If this field is set, the specified message queue must exist on all nodes in the cluster when they are started. The message queue cannot be in an independent disk pool.

- In the **Library** field, specify the name of the library that contains the message queue to receive the failover message. The library name cannot be *CURLIB, QTEMP, *LIBL, *USRLIBL, *ALL, or *ALLUSR.
- In **Failover wait time** field, select either **Do not wait** or **Wait forever**, or specify the number of minutes to wait for a reply to the failover message on the cluster message queue.
- In the **Failover default action** field, specify the action that Cluster Resource Services takes when the response to the failover message has exceeded the failover wait time value. You can set this field to **Proceed with failover** or to **Cancel failover**.

Specifying a failover message queue

To specify a failover message queue, complete these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from your IBM Systems Director Navigator for i5/OS window.
4. On the Cluster Resource Services page, select **Work with Cluster Resource Groups** to show a list of cluster resource groups in the cluster.
5. From the list of cluster resource groups, select the cluster resource group with which you want to work.
6. On the Cluster Resource Group page, click the **Select Action** menu and select **Properties**.
7. On the General page, specify the following values to specify a failover message queue:
   - In the **Failover message queue** field, specify the name of the message queue to receive messages when a failover occurs for this cluster resource group. If this field is set, the specified message queue must exist on all nodes in the recovery domain after the exit program is completed. The failover message queue cannot be in an independent disk pool.
   - In the **Library** field, specify the name of the library that contains the message queue to receive the failover message. The library name cannot be *CURLIB, QTEMP, or *LIBL.
   - In the **Failover wait time** field, specify the number of minutes to wait for a reply to the failover message on the failover message queue. You can also specify the action that Cluster Resource Services takes when a response to the failover message exceeds the specified failover wait time.

Cluster deconfiguration checklist

To ensure complete deconfiguration of a cluster, you must systematically remove different cluster components.

Table 9. Independent disk pool deconfiguration checklist for clusters

Independent disk pool requirements	
__	If you are using switched disk pools, the tower should be switched to the node which is the SPCN owner before deconfiguring the cluster resource group. You can use Initiate Switchover (QcstInitiateSwitchOver) API or the Change Cluster Resource Group Primary (CHGCRGPRI) command to move the CRG back to the SPCN owner. If this step is not performed, you will not be able to mark the tower private for that system.
__	If you plan to remove a subset of an independent disk pool group or remove the last independent disk pool in the switchable devices, you must end the CRG first. Use the End Cluster Resource Group (ENDCRG) command.

Table 9. Independent disk pool deconfiguration checklist for clusters (continued)

Independent disk pool requirements	
__	If you want delete an independent disk pool that is participating in a cluster, it is strongly recommended that you first delete the device cluster resource group (CRG). See "Deleting a CRG" on page 140 for details. You can also use the **Remove CRG Device Entry (RMVCRGDEVE)** command to remove the configuration object of the independent disk pool from the CRG.
__	After you have removed the configuration object of the independent disk pool from the cluster switchable device, you can delete an independent disk pool.
__	Delete the device description for an independent disk pool by completing these tasks: 1. On a command-line interface, type **WRKDEVD DEVD(*ASP)** and press Enter. 2. Page down until you see the device description for the independent disk pool that you want to delete. 3. Select Option 4 (Delete) by the name of the device description and press Enter.

Table 10. Cluster resource group deconfiguration checklist for clusters

Cluster resource group requirement	
__	Delete cluster resource group by completing the either of the following steps: 1. If clustering is not active on the node, then type **DLTCRG CRG(CRGNAME)** on a command-line interface. CRGNAME is the name of the CRG that you want to delete. Press Enter. 2. If clustering is active on the node, then type **DLTCRGCLU CLUSTER(CLUSTERNAME) CRG(CRGNAME)** on a command-line interface. CLUSTERNAME is the name of the cluster. CRGNAME is the name of the CRG that you want to delete. Press Enter.

Managing nodes

System and logical partitions that are a part of an i5/OS high availability environment are called nodes. You can perform several managing tasks that pertain to nodes.

Displaying node properties

You can display and manage properties that are associated with nodes that are configured as part of your high-availability environment by using the Cluster Resource Services graphical interface.

To display node properties, complete the following tasks

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the **Cluster Resource Services** page, select the **Work with Cluster Nodes** task to show a list of nodes in the cluster.
5. On the **Nodes** tab, click the **Select Action** menu and select **Properties**. Click **Go**. This displays the Node properties page.
   - The General page displays the name of the node and the system IP address for that node.
   - The Clustering page displays the following information:
     - The cluster interface IP addresses that are used by clustering to communicate with other nodes in the cluster.
     - The potential version of the node specifies the version and modification level at which the nodes in the cluster are actively communicating with each other.
     - The device domains that are configured in the selected cluster. If you select a device domain in the list, the nodes that belong to the selected device domain are also displayed.

Stopping nodes

Stopping or ending a node ends clustering and cluster resource services on that node.

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the **Nodes** tab, select the node you want to stop.
5. Click the **Select Action** menu and click **Stop**. When cluster resource services is successfully stopped on the node specified, the status of the node is set to Stopped.

Related information:

End Cluster Node (ENDCLUNOD) command

End Cluster Node (QcstEndClusterNode) API

Removing nodes

You might need to remove a node from a cluster if you are performing an upgrade of that node or if the node no longer needs to participate in the i5/OS high-availability environment.

To remove a node from an existing cluster, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the **Cluster Resource Services** page, select the **Work with Cluster Nodes** task to show a list of nodes in the cluster.
5. On the Nodes page, select the **Select Action** menu, and then select **Remove**.
6. Click **Yes** in the Remove Cluster Node Confirmation window.

Related tasks:

"Deconfiguring geographic mirroring" on page 174

If you no longer want the capability to use geographic mirroring for a specific disk pool or disk pool group, you can select to **Deconfigure Geographic Mirroring**. If you deconfigure geographic mirroring, the system stops geographic mirroring and deletes the mirror copy of the disk pools on the nodes in the mirror copy site.

Related information:

Remove Cluster Node Entry (RMVCLUNODE) command

Remove Cluster Node Entry (QcstRemoveClusterNodeEntry) API

Removing a node from a device domain

A *device domain* is a subset of nodes in a cluster that share device resources.

Important:

Be cautious when removing a node from a device domain. If you remove a node from a device domain, and that node is the current primary point of access for any independent disk pools, those independent disk pools remain with the node being removed. This means that those independent disk pools are no longer accessible from the remaining nodes in the device domain.

After a node is removed from a device domain, it cannot be added back to the same device domain if one or more of the existing cluster nodes still belong to that same device domain. To add the node back to the device domain, you must:

1. Delete the independent disk pools currently owned by the node being added to the device domain.
2. Restart the system by performing an IPL on the node.

3. Add the node to the device domain.
4. Re-create the independent disk pools deleted in step 1.

To remove a node from a device domain, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the **Cluster Resource Services** page, select the **Work with Cluster Nodes** task to show a list of nodes in the cluster.
5. On the **Nodes** tab, select the **Select Action** menu and select **Properties**. Click **Go**. The Node Properties sheet is shown.
6. On the **Clustering** tab, delete the node name from the **Device domain** field and click **OK**.

Related tasks:

"Deleting a cluster" on page 130
When you delete a cluster, cluster resource services ends on all active cluster nodes and they will be removed from the cluster.

Related information:

Remove Device Domain Entry (RMVDEVDMNE) command

Remove Device Domain Entry (QcstRemoveDeviceDomainEntry) API

Add a cluster monitor to a node

IBM i Cluster Resource Services can now use Hardware Management Console (HMC) or a Virtual I/O Server (VIOS) partition to detect when a cluster node fails. This new capability allows more failure scenarios to be positively identified and avoids cluster partition situations.

The PowerHA graphical interface allows you to use the HMC or VIOS to monitor and manage the state of each system. Once a monitor is set up, HMC or VIOS provide notification of node failures. A cluster monitor can be used to reduce the number of failure scenarios which result in cluster partitions.

PowerHA GUI only supports the Common Informational Model (CIM) server for the cluster monitor. The Add Cluster Monitor command must be used if you want to use the representational state transfer (REST) server.

To add a cluster monitor to an existing cluster, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the **Cluster Resource Services** page, select the **Work with Cluster Nodes** task to show a list of nodes in the cluster.
5. On the **Nodes** tab, click the context icon next to the specific node, and then **Select Properties**. The Node Properties sheet is shown.
6. On the **Monitors** tab, on the monitor information table click the **Select Action** drop-down list and select the **Add Cluster Monitor** action.

Removing a cluster monitor

A *cluster monitor* provides another source of information to allow cluster resource services to determine when a cluster node has failed.

Important:

Be cautious when removing a cluster monitor. If you remove a node from a cluster monitor, and that node is the current primary point of access for any CRG, that node could partition when in fact, the node really failed. This means the user must now do manual steps to become highly available again.

PowerHA GUI only supports the Common Informational Model (CIM) server for the cluster monitor. The Remove Cluster Monitor command must be used if you want to use the representational state transfer (REST) server.

To remove a cluster monitor, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the **Cluster Resource Services** page, select the **Work with Cluster Nodes** task to show a list of nodes in the cluster.
5. On the **Nodes** tab, click the context icon next to the specific node, and then select **Properties**. Click **Go**. The Node Properties sheet is shown.
6. Select the **Monitors** tab, to see a list of cluster monitors configured for the node.
7. On the **Monitors** tab, select the specific monitor, click the **Select action** drop-down list, and select the **Remove** function.

Chapter 6. Managing cluster resource groups (CRGs)

Cluster resource groups (CRGs) manage resilient resources within an i5/OS high availability environment. They are a cluster technology that defines and controls switching resources to backup systems in the event of an outage.

Displaying CRG status

You can monitor the status of cluster resource groups (CRG) in your high-availability environment. You can use these status messages to validate changes in the CRG or to determine problems with the CRG.

To display the CRG status, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, select **Work with Cluster Resource Groups** to show a list of cluster resource groups in the cluster.
5. On the Cluster Resource Group page, view the current status of a CRG in the Status column.

The following are possible status values for a CRG:

Table 11. Status values for CRGs

Possible values	Description
Started	The CRG is currently started.
Stopped	The CRG is currently stopped.
Indoubt	The information about this CRG within the high availability solution might not be accurate. This status occurs when the CRG exit program is called with an action of undo and fails to complete successfully.
Restored	The CRG was restored on its node and has not been copied to other nodes in the cluster. When clustering is started on the node, the CRG will be synchronized with the other nodes and its status is set to inactive.
Inactive	Cluster resource services for the CRG is not active on the node. The node might have failed, the node might have been ended, or the CRG job on that node might not be running.
Deleting	The CRG is in the process of being deleted from the cluster.
Changing	The CRG is in the process of being changed. The CRG is reset to its previous status when the change has been successfully completed.
Stopping	The CRG is in the process of being stopped.
Adding	The CRG is in the process of being added to the cluster.
Starting	The CRG is in the process of being started.
Switching	The CRG is in the process of switching over to another node.

Table 11. Status values for CRGs (continued)

Possible values	Description
Adding node	A new node is in the process of being added to the cluster. The CRG is reset to its previous status when the node has been successfully added.
Removing node	A node is in the process of being removed from the CRG. The CRG is reset to its previous status when the node has been successfully removed.
Changing node status	The status of a node in the recovery domain for a CRG is currently being changed.

Stopping a CRG

Cluster resource groups (CRGs) manage resilient resources within an i5/OS high availability environment. They are a cluster technology that defines and controls switching resilient resources to backup systems in the event of an outage.

You might want to stop the CRG to end automatic failover capability in your high-availability environment. For example, you might be performing an IPL on one of the systems that is defined in the CRG.

To stop a CRG, complete these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, select **Work with Cluster Resource Groups** to show a list of cluster resource groups in the cluster.
5. On the Cluster Resource Group page, select a CRG that you want to stop.
6. From the **Select Action** menu, select **Stop** and click **Go**.

Related information:

End Cluster Resource Group (ENDCRG) command

End Cluster Resource Group (QcstEndClusterResourceGroup) API

Deleting a CRG

You can delete a cluster resource group by using the Cluster Resource Service interface.

To delete a CRG, complete these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, select **Work with Cluster Resource Groups** to show a list of cluster resource groups in the cluster.
5. On the Cluster Resource Group page, select a CRG that you want to delete.
6. From the **Select Action** menu, select **Delete** and click **Go**.
7. Select **Yes** in the Delete Cluster Resource Group confirmation window.

Related information:

Delete Cluster Resource Group from Cluster (DLTCRGCLU) command

Delete Cluster Resource Group (QcstDeleteClusterResourceGroup) API

Creating switchable devices

In addition to independent disk pool devices, several other devices are supported for high availability. Devices, such as Ethernet lines, optical devices, and network servers, and others, can now be part of a high availability solution.

A device cluster resource group contains a list of switchable devices. Each device in the list identifies a switchable independent disk pool or another type of switchable device, such as tape devices, line descriptions, controllers, and network servers. The entire collection of devices is switched to the backup node when an outage occurs. You also can vary on the devices during the switchover or failover process.

To create a switchable device, complete the following steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, select **Work with Cluster Resource Groups** to show a list of cluster resource groups in the cluster.
5. On the Cluster Resource Group page, click the context icon next to the device cluster resource group for which you would like to add an existing switchable device, and select **Add Existing Device** from the context menu.
6. In the Add Switchable Device List, click **Add**.
7. In the Add Switchable Device window, fill in the configuration object type and object name of the switchable device. Click **OK** to add the new switchable device to the list. For example, if you were adding a switchable Ethernet line, select Ethernet line for the list.
8. Click **OK** on the list window to add the new device to the device cluster resource group.

Changing the recovery domain for a CRG

The recovery domain controls recovery actions for a subset of nodes defined in a cluster resource group (CRG).

To change a recovery domain for a device cluster resource group, application cluster resource group, or data cluster resource group, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, select **Work with Cluster Resource Groups** to show a list of cluster resource groups in the cluster.
5. On the Cluster Resource Group page, select a CRG that you want to change.
6. From the **Select Action** menu, select **Properties** and click **Go**.
7. Click the Recovery Domain page to change the existing values for the recovery domain. On this page you can change the roles of nodes within the recovery domain of the cluster, and add and remove nodes from the recovery domain. For a device cluster resource group, you can also change the site name and data port IP addresses for a node in the recovery domain.

Related information:

Add Cluster Resource Group Node Entry (ADDCRGNODE) command

Change Cluster Resource Group (CHGCRG) command

Remove Cluster Resource Group Node Entry (RMVCRGNODE) command

Add a Node to Recovery Domain (QcstAddNodeToRcvyDomain) API

Change Cluster Resource Group (QcstChangeClusterResourceGroup) API

Creating site names and data port IP addresses

If you are using geographic mirroring, the nodes defined in the recovery domain node of the device cluster resource group must have a data port IP address and site name.

The site name is associated with a node in the recovery domain for a device cluster resource group, applicable only to geographic mirroring. When you are configuring a geographic mirroring environment for high availability, each node at different sites must be assigned to a different site name.

To create the data port IP address and site names for nodes in the recovery domain, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, click the **Work with Cluster Resource Groups** task to show a list of cluster resource groups in the cluster.
5. On the Cluster Resource Group tab, click the context icon next to the device cluster resource group, and then select **Properties**.
6. On the Recovery Domain page, select **Edit**.
7. To use an existing data port IP address, select it from the list and click **OK**. To add a new data port IP address, click **Add**. In the Add Data Port IP Address window, enter the IP address.
8. In the Edit window, you can specify the Site name.

Chapter 7. Managing failover outage events

Typically, a failover results from a node outage, but there are other reasons that can also generate a failover. Different system or user actions can potentially cause failover situations.

It is possible for a problem to affect only a single cluster resource group (CRG) that can cause a failover for that CRG but not for any other CRG.

Four categories of outages can occur within a cluster. Some of these events are true failover situations where the node is experiencing an outage, while others require investigation to determine the cause and the appropriate response. The following tables describe each of these categories of outages, the types of outage events that fall into that category and the appropriate recovery action you should take to recover.

Category 1 outages: Node outage causing failover

Node-level failover occurs, causing the following to happen:

- For each CRG, the primary node is marked *inactive* and made the last backup node.
- The node that was the first backup becomes the new primary node.

Failovers happen in this order:

1. All device CRGs
2. All data CRGs
3. All application CRGs

Notes:

1. If a failover for any CRG detects that none of the backup nodes are active, the status of the CRG is set to *indoubt* and the CRG recovery domain does not change.
2. If all of cluster resource services fails, then the resources (CRGs) that are managed by cluster resource services go through the failover process.

Table 12. Category 1 outages: Node outage causing failover

Failover outage event
ENDTCP(*IMMED or *CNTRLD with a time limit) is issued.
ENDSYS (*IMMED or *CNTRLD) is issued.
PWRDWNSYS(*IMMED or *CNTRLD) is issued.
Initial program load (IPL) button is pressed while cluster resource services is active on the system.
End Cluster Node (API or command) is called on the primary node in the CRG recovery domain.
Remove Cluster Node (API or command) is called on the primary node in the CRG recovery domain.
HMC delayed power down of the partition or panel option 7 is issued.
ENDSBS QSYSWRK(*IMMED or *CNTRLD) is issued.

Category 2 outages: Node outage causing partition or failover

These outages will cause either a partition or a failover depending on whether advanced node failure detection is configured. Refer to the columns in the table. If advanced node failure detection is configured, failover occurs in most cases and Category 1 outage information applies. If advanced node failure detection is not configured, partition occurs and the following applies:

- The status of the nodes not communicating by cluster messaging is set to a Partition status. See Cluster partition for information about partitions.
- All nodes in the cluster partition that do not have the primary node as a member of the partition will end the active cluster resource group.

Notes:

1. If a node really failed but is detected only as a partition problem and the failed node was the primary node, you lose all the data and application services on that node and no automatic failover is started.
2. You must either declare the node as failed or bring the node back up and start clustering on that node again. See Change partitioned nodes to failed for more information.

Table 13. Category 2 outages: Node outage causing partition

Failover outage event	No advanced node failure detection	HMC	VIOS
CEC hardware outage (CPU, for example) occurs.	partition	failover	partition or failover
Operating system software machine check occurs.	partition	failover	failover
HMC immediate power off or panel option 8 is issued.	partition	failover	failover
HMC partition restart or panel option 3 is issued.	partition	failover	failover
Power loss to the CEC occurs.	partition	partition	partition

Category 3 outages: CRG fault causing failover

For a system containing VIOS, a CEC hardware failure could result in either failover or partition. Which occurs depends upon the type of system and the hardware failure. For example in a blade system, a CEC failure that prevents VIOS from running results in a partition since VIOS is unable to report any failure. In the same system in which a single blade fails but VIOS continues to run, failover results since VIOS is able to report the failure.

When a CRG fault causes a failover, the following happens:

- If only a single CRG is affected, failover occurs on an individual CRG basis. This is because CRGs are independent of each other.
- If someone cancels several cluster resource jobs, so that several CRGs are affected at the same time, no coordinated failover between CRGs is performed.
- The primary node is marked as Inactive in each CRG and made the last backup node.
- The node that was the first backup node becomes the new primary node.
- If there is no active backup node, the status of the CRG is set to Indoubt and the recovery domain remains unchanged.

Table 14. Category 3 outages: CRG fault causing failover

Failover outage event
The CRG job has a software error that causes it to end abnormally.
Application exit program failure for an application CRG.

Category 4 outages: Communication outage causing partition

This category is similar to category 2. These events occur:

- The status of the nodes not communicating by cluster messaging are set to Partition status. See Cluster partition for information about partitions.
- All nodes and cluster resource services on the nodes are still operational, but not all nodes can communicate with each other.
- The cluster is partitioned, but each CRG's primary node is still providing services.

The normal recovery for this partition state should be to repair the communication problem that caused the cluster partition. The cluster will resolve the partition state without any additional intervention.

Note: If you want the CRGs to fail over to a new primary node, ensure that the old primary node is not using the resources before the node is marked as failed. See Change partitioned nodes to failed for more information.

Table 15. Category 4 outages: Communication outage causing partition

Failover outage event
Communications adapter, line, or router failure on cluster heartbeat IP address lines occurs.
ENDTCPIFC is affecting all cluster heartbeat IP addresses on a cluster node.

Outages with active CRGs

- If the CRG is Active and the failing node is *not* the primary node, the following results:
  - The failover updates the status of the failed recovery domain member in the CRG's recovery domain.
  - If the failing node is a backup node, the list of backup nodes is reordered so that active nodes are at the beginning of the list.
- If the CRG is Active and the recovery domain member is the primary node, the actions the system performs depend on which type of outage has occurred.
  - Category 1 outages: Node outage causing failover
  - Category 2 outages: Node outage causing partition
  - Category 3 outages: CRG fault causing failover
  - Category 4 outages: Communication outage causing partition

Outages with inactive CRGs

When there is an outage with CRGs, the following occur:

- The membership status of the failed node in the cluster resource group's recovery domain is changed to either Inactive or Partition status.
- The node roles are not changed, and the backup nodes are not reordered automatically.
- The backup nodes are reordered in an Inactive CRG when the **Start Cluster Resource Group (STRCRG)** command or the Start Cluster Resource Group (QcstStartClusterResourceGroup) API is called.

  Note: The Start Cluster Resource Group API will fail if the primary node is not active. You must issue the **Change Cluster Resource Group (CHGCRG)** command or the Change Cluster Resource Group (QcstChangeClusterResourceGroup) API to designate an active node as the primary node, and then call the Start Cluster Resource Group API again.

Chapter 8. Managing cluster administrative domains

After a cluster administrative domain is created and the appropriate monitored resource entries (MREs) are added, the cluster administrator should monitor the activity within the administrative domain to ensure that the monitored resources remain consistent. Using Cluster Resource Services graphical interface, you can manage and monitor a cluster administrative domain.

This graphical interface provides the ability to list the MREs along with the global status for each resource. Detailed information can be displayed by selecting an MRE. This information includes the global value for each attribute associated with the MRE, along with an indication whether the attribute is consistent or inconsistent with the domain. If the global status of a monitored resource is inconsistent, the administrator should take the necessary steps to determine why the resource is inconsistent, correct the problem, and resynchronize the resource.

If the resource is inconsistent because an update failed on one or more nodes, information is kept for the MRE that can help you determine the cause of the failure. On the node where the failure occurred, a message is logged with the MRE as to the cause of the failed update. On the other nodes, there will be an informational message logged internally which tells you there was a failure, along with the list of nodes where the update failed. These messages are available through the Cluster Resource Service graphical interface or by calling the Retrieve Monitored Resource Information (QfpadRtvMonitoredResourceInfo) API. Failure messages are also logged in the job log of the peer CRG job.

After the cause of the inconsistency is determined, the resource can be resynchronized, either as a result of an update operation on the node where the failure occurred, or by ending and restarting the administrative domain. For example, an MRE for a user profile could be inconsistent because you changed the UID for the user profile on one node in the administrative domain, but the UID you specified was already in use by another user profile on one of the nodes. If you change the value of the UID again to something that is not used by another user profile within the administrative domain, the change will be made by the cluster administrative domain on all nodes and the global status for the user profile MRE is set to consistent. You do not need to take any further action to resynchronize the user profile MRE.

In some cases, you need to end and restart the cluster administrative domain CRG in order for the inconsistent resources to be resynchronized. For example, if you change the UID for a user profile that has an MRE associated with it, but the user profile is active in a job on one of the other cluster nodes in the administrative domain, the global value for the MRE associated with the user profile will be set to inconsistent because the change operation failed on the node where the user profile was active in a job. In order to correct this situation, you need to wait until the job has ended and then end the cluster administrative domain. When the administrative domain is started again, the global value for each attribute that is inconsistent will be used to change the resource to a consistent state.

The global status for a monitored resource is always set to failed if the resource is deleted, renamed, or moved on any node in the domain. If this is the case, the MRE should be removed because the resource is no longer be synchronized by the cluster administrative domain.

When you restore a monitored resource on any system that is part of a cluster administrative domain, the resource is resynchronized to the global value currently known in the cluster administrative domain when the peer CRG representing the cluster administrative domain is active.

The following restore commands result in a resynchronization of system objects: RSTLIB, RSTOBJ, RSTUSRPRF and RSTCFG. In addition, RSTSYSINF and UPDSYSINF result in a resynchronization of

system values and network attributes. To resynchronize system environment variables after running the RSTSYSINF or UPDSYSINF commands, the peer CRG that represents the cluster administrative domain must be ended and started again.

If you want to restore your monitored resources to a previous state, remove the MRE that represents the resource that you want to restore. Then, after restoring the resource, add an MRE for the resource from the system where the restore operation was done. The cluster administrative domain will synchronize the monitored resource across the domain by using the values from the restored resource.

To monitor a cluster administrative domain, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. On the Cluster Resource Services page, click **Work with Administrative Domains** to display a list of cluster administrative domains in the cluster.
4. On the **Administrative Domain** tab, select **New Administrative Domain**.
5. On the New Administrative Domain page, specify the information on the cluster administrative domain.

Stopping a cluster administrative domain

Cluster administrative domains provide environment resiliency for resources within an i5/OS high-availability solution. You might need to stop a cluster administrative domain to temporarily end synchronization of monitored resources.

A cluster administrative domain becomes inactive when it is stopped. While the cluster administrative domain is inactive, all of the monitored resources are considered to be inconsistent because changes to them are not being synchronized. Although changes to monitored resources continue to be tracked, the global value is not changed and changes are not propagated to the rest of the administrative domain. Any changes made to any monitored resource while the cluster administrative domain is inactive are synchronized across all active nodes when the cluster administrative domain is restarted.

Note: The cluster administrative domain and its associated exit program are IBM-supplied objects. They should not be changed using the QcstChangeClusterResourceGroup API or the **CHGCRG** command. Making these changes will cause unpredictable results.

To stop a cluster administrative domain, complete these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. On the Cluster Resource Services page, click **Work with Administrative Domains** to display a list of cluster administrative domains in the cluster.
4. On the Administrative Domains page, select a cluster administrative domain.
5. From the **Select Action** menu, select **Stop**.
6. Click **Yes** on the Stop Administrative Domain Confirmation page.

Related information:

End Cluster Administrative Domain (ENDCAD) command

Deleting a cluster administrative domain

Using the Cluster Resource Services interface, you can delete a cluster administrative domain. Deleting a cluster administrative domain ends synchronization of monitored resources that are defined in the cluster administrative domain.

To delete a cluster administrative domain, complete the following:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. On the Cluster Resource Services page, click **Work with Administrative Domains** to display a list of cluster administrative domains in the cluster.
4. On the Administrative Domains page, select a cluster administrative domain.
5. From the **Select Action** menu, select **Delete**.
6. Click **Yes** on the Delete Administrative Domain Confirmation page.

Changing the properties of a cluster administrative domain

Using the Cluster Resource Services graphical interface, you can change properties to an existing cluster administrative domain. These properties control synchronization of monitored resource entries that are defined in the cluster administrative domain.

To change the properties of a cluster administrative domain, complete the following steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. On the Cluster Resource Services page, click **Work with Administrative Domains** to display a list of cluster administrative domains in the cluster.
4. On the Administrative Domains page, select a cluster administrative domain.
5. From the **Select Action** menu, select **Properties**.
6. On the Properties page, you can change the following information about the cluster administrative domain:
   - In the **Name** field, enter the name of the cluster administrative domain. The name cannot exceed 10 characters.
   - In the **Synchronization option** field, specify the synchronization behavior when a node joins a cluster administrative domain. This field is enabled only if the cluster is at version 6 or greater. Possible values follow:

     Last Change Option (default)
     Select this option if all changes to monitored resources should be applied to a cluster administrative domain. When a node joins an active cluster administrative domain, any changes made to monitored resources on the joining node while it was inactive, are applied to the monitored resources on the other active nodes in the domain, unless a more recent change was made to the resource in the active domain. The most recent change that is made to a monitored resource is applied to the resource on all active nodes.

     Active Domain Option
     Select this option if only changes to monitored resources are allowed from active nodes. Changes made to monitored resources on inactive nodes are discarded when the node joins the cluster administrative domain. The Active Domain option does not apply to network server storage spaces (*NWSSTG) or network server configurations (*NWSCFG). Synchronization of these resources is always based on the last change that was made.
   - From the **Nodes in the administrative domain** list, you can either add a node to the cluster administrative domain by selecting **Add** or you can remove a node from the domain by selecting **Remove**.

Managing monitored resource entries

The Cluster Resource Services graphical interfaces allows you to manage monitored resource entries in your cluster administrative domain. A cluster administrative domain ensures that changes made to these monitored resources remain consistent on each node within the high-availability environment.

Working with monitored resource entry status

The Cluster Resource Services graphical interface provides status messages for monitored resource entries within a cluster administrative domain.

After an MRE is added to the cluster administrative domain, the resource is monitored for changes on all administrative domain nodes so that the values of the resource attributes can be synchronized across the nodes in the cluster administrative domain. The synchronization behavior is dependent on a number of factors:

- Status of the cluster
- Status of the cluster administrative domain
- Status of the node
- Particular actions on the resource

To work with monitored resource entry status, complete these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, click **Work with Administrative Domains** to display a list of cluster administrative domains in the cluster.
5. On the Administrative Domains page, click the context icon next to the cluster administrative domain name, and select **Monitored Resource Entries**.

   Note: The **Monitored Resource Entries** action is available only if the node that you are managing is part of the cluster administrative domain. The current list of monitored resource types is shown.
6. On the **Monitored Resources Entries** list panel, click the context icon next to the resource type and select **Attributes**.
7. The list of attributes for monitored resource is shown. The Global Status column displays the current status for this attribute in the active cluster administrative domain.

   These values determine the status of a monitored resource across the cluster:

   Global Value
   The value for each monitored attribute that a resource is expected to have on all administrative domain nodes. The global value is the same on all active nodes and represents the last change that was synchronized in the domain.

   Global Status
   The status of the resources across a cluster administrative domain and whether the resources are fully synchronized. Possible global status values follow:

   Consistent
   The values for all the resource's attributes monitored by the system are the same on all active nodes within the cluster administrative domain. This status occurs in a normal operational environment where the cluster, the cluster administrative domain, and all of the nodes are operational and active in the cluster. In this environment, any change to a value of a monitored resource is propagated to all the other nodes in the cluster administrative domain. This processing is asynchronous to the original change, but will result in consistent values for the enrolled resources across the administrative domain. In this situation, the global status is Consistent, the change is successfully made on each node, and the value of the resource on each node matches the global value for the resource.

   Inconsistent
   The values for all the resource's attributes monitored by the system are not the same on all active nodes within the cluster administrative domain. A message is logged that describes why the status is Inconsistent. For example, if changes were made to

monitored resources while the cluster administrative domain is inactive, then monitored resource status would be Inconsistent.

Pending
The values of the monitored attributes are in the process of being synchronized across the cluster administrative domain.

Added
The monitored resource entry has been added to the cluster administrative domain but has not yet been synchronized.

Ended
The monitored resource is in an unknown state because the cluster administrative domain has been ended, and changes to the resource are no longer being processed. When the cluster administrative domain is ended, the global status for all MREs that are currently set to Consistent are set to Ended.

Failed
The resource is no longer being monitored by the cluster administrative domain and the MRE should be removed. Certain resource actions are not recommended when a resource is being synchronized by a cluster administrative domain. If the resource represented by an MRE is a system object, it should not be deleted, renamed, or moved to a different library without removing the MRE first. If a resource is deleted, renamed or moved to a different library, the global status for the MRE is Failed and any changes made to the resource on any node after that are not propagated to any node in the cluster administrative domain.

When restoring a monitored resource on a node within the cluster administrative domain, the values of the monitored resource are changed back to match the global values that are synchronized by the cluster administrative domain.

Removing monitored resource entries

Monitored resource entries (MREs) are resources currently used within the high-availability environment and are monitored for changes through a cluster administrative domain. You might want to remove MREs when you no longer need them to be monitored. You can remove monitored resource entries (MREs) by using the Cluster Resource Services graphical interface.

To remove a monitored resource entry, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, click **Work with Administrative Domains** to display a list of cluster administrative domains in the cluster.
5. On the Administrative Domains page, click the context icon next to the cluster administrative domain name, and select **Monitored Resource Entries**.

   Note: The **Monitored Resource Entries** action is available only if the node that you are managing is part of the cluster administrative domain. The current list of monitored resource types is shown.
6. In the list of monitored resource types, click the context icon next to the monitored resource type, and select **Monitored Resource Entries**. The MRE object list is shown.
7. Click the context icon next to the MRE object that you would like to remove, and select **Remove Monitored Resource Entry**.
8. Click **Yes** in the Remove MRE Object Confirmation window. The monitored resource entry is removed from the cluster administrative domain.

Related information:

Remove Admin Domain MRE (RMVCADMRE) command

Listing monitored resource entries

Monitored resource entries are resources, such as user profiles and environment variables, that have been defined in a cluster administrative domain. You can use the Cluster Resource Services graphical interface to list monitored resource entries that are currently defined in a cluster administrative domain.

To list monitored resource entries, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, click **Work with Administrative Domains** to display a list of cluster administrative domains in the cluster.
5. On the Administrative Domains page, click the context icon next to the cluster administrative domain name, and select **Monitored Resource Entries**.

   Note: The **Monitored Resource Entries** action is available only if the node that you are managing is part of the cluster administrative domain. The current list of monitored resource types is shown.
6. In the list of monitored resource types, click the context icon next to the monitored resource type, and select **Monitored Resource Entries**.
7. View and work with the list of enrolled monitored resource entries.

Selecting attributes to monitor

After you have added monitored resource entries, you can select attributes associated with that resource to be monitored by the cluster administrative domain.

To select attributes to monitor for a monitored resource entry (MRE), follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, click **Work with Administrative Domains** to display a list of cluster administrative domains in the cluster.
5. On the Administrative Domains page, click the context icon next to the cluster administrative domain name, and select **Monitored Resource Entries**.

   Note: The **Monitored Resource Entries** action is available only if the node that you are managing is part of the cluster administrative domain. The current list of monitored resource types is shown.
6. In the list of monitored resource types, click the context icon next to the monitored resource type, and select **Monitored Resource Entries...**. The MRE object list is shown.
7. Click the context icon next to the MRE object, such as user profile or system value, and select **Work with Attributes**. The MRE Attributes List is shown.
8. In the MRE Attribute List window, select the attributes that you want to monitor, and then click **Close**. For example, if you want to monitor Ethernet line description for changes to its resource name attribute, you would select resource name as the attribute.

Related tasks:

"Adding monitored resource entries" on page 113
You can add a monitored resource entry (MRE) to a cluster administrative domain. Monitored resource entries define critical resources so that changes made to these resources are kept consistent across a high-availability environment.

Attributes that can be monitored

A monitored resource entry can be added to the cluster administrative domain for various types of resources. This topic lists the attributes that each resource type can be monitored.

Resource types

- Authorization lists (*AUTL)
- Classes (*CLS)
- Ethernet line descriptions (*ETHLIN)
- Independent disk pools device descriptions (*ASPDEV)
- Job descriptions (*JOBD)
- Network attributes (*NETA)
- Network server configuration for connection security (*NWSCFG)
- Network server configuration for remote systems (*NWSCFG)
- Network server configurations for service processors (*NWSCFG)
- Network server descriptions for iSCSI connections (*NWSD)
- Network server descriptions for integrated network servers (*NWSD)
- Network server storage spaces (*NWSSTG)
- Network server host adapter device descriptions (*NWSHDEV)
- Optical device descriptions (*OPTDEV)
- Printer device descriptions for LAN connections (*PRTDEV)
- Printer device descriptions for virtual connections (*PRTDEV)
- Subsystem descriptions (*SBSD)
- System environment variables (*ENVVAR)
- System values (*SYSVAL)
- Tape device descriptions (*TAPDEV)
- Token-ring line descriptions (*TRNLIN)
- TCP/IP attributes (*TCPA)
- User profiles (*USRPRF)

Table 16. Table 1. Attributes that can be monitored for authorization lists

Attribute name	Description
AUT	Authority
TEXT	Text description

Table 17. Attributes that can be monitored for classes

Attribute name	Description
CPUTIME	Maximum CPU time
DFTWAIT	Default wait time
MAXTHD	Maximum threads
MAXTMPSTG	Maximum temporary storage
RUNPTY	Run priority
TEXT	Text description
TIMESLICE	Time slice

Table 18. Attributes that can be monitored for Ethernet line descriptions

Attribute name	Description
ASSOCPORT	Associated port resource name
AUTOCRTCTL	Autocreate controller
AUTODLTCTL	Autodelete controller
CMNRCYLMT	Recovery limits
COSTBYTE	Relative cost per byte for sending and receiving data on the line
COSTCNN	Relative cost of being connected on the line
DUPLEX	Duplex
GENTSTFRM	Generate test frames
GRPADR	Group address
LINESPEED	Line speed
MAXFRAME	Maximum frame size
MAXCTL	Maximum controllers
MSGQ	Message queue
ONLINE	Online at IPL
PRPDLY	Propagation delay
RSRCNAME	Resource name
SECURITY	Security level of the physical line
SSAP	Source service access point (SSAP) information list
TEXT	Text description
USRDFN1	First user-defined
USRDFN2	Second user-defined
USRDFN3	Third user-defined
VRYWAIT	Vary on wait

Table 19. Attributes that can be monitored for independent disk pools device descriptions

Attribute name	Description
MSGQ	Message queue
RDB	Relational database
RSRCNAME	Resource name
TEXT	Text description

Table 20. Attributes that can be monitored for job descriptions

Attribute name	Description
ACGCDE	Accounting code
ALWMLTTHD	Allow multiple threads
DDMCNV	DDM conversation
DEVRCYACN	Device recovery action
ENDSEV	End severity
HOLD	Hold on job queue

Table 20. Attributes that can be monitored for job descriptions (continued)

Attribute name	Description
INLASPGRP	Initial ASP group
INLLIBL	Initial library list
INQMSGRPY	Inquiry message reply
JOBMSGQFL	Job message queue full action
JOBMSGQMX	Job message queue maximum size
JOBPTY	Job priority (on JOBQ)
JOBQ	Job queue
LOG	Message logging
LOGCLPGM	Log CL program commands
OUTPTY	Output priority (on OUTQ)
OUTQ	Output queue
PRTDEV	Print device
PRTTXT	Print text
RQSDTA	Request data or command
RTGDTA	Routing data
SPLFACN	Spooled file action
SWS	Job switches
SYNTAX	CL syntax check
TEXT	Text description
TSEPOOL	Time slice end pool
USER	User

Table 21. Attributes that can be monitored for network attributes

Attribute name	Description
ALWADDCLU	Allow add to cluster
DDMACC	DDM/DRDA request access
NWSDOMAIN	Network server domain
PCSACC	Client request access
Note: Each network attribute is treated as its own monitored resource entry. For these, the resource type and attribute names are identical.	

Table 22. Attributes that can be monitored for network server configurations for service processors

Attribute name	Description
EID	Enclosure identifier
INZSP	Initialize service processor
SPAUT	Service processor authority
SPCERTID	Service processor certificate identifier
SPINTNETA	Service processor Internet address
SPNAME	Service processor name
TEXT	Text description

Table 23. Attributes that can be monitored for network server configuration for remote systems

Attribute name	Description
BOOTDEVID	Boot device identifier
CHAPAUT	Target CHAP authentication
DELIVERY	Delivery method
DYNBOOTOPT	Dynamic boot options
INRCHAPAUT	Initiator CHAP authentication
RMTIFC	Remote interfaces
RMTSYSID	Remote system identifier
SPNWSCFG	Service processor network server configuration that is used to manage the remote server
TEXT	Text description

Table 24. Attributes that can be monitored for network server configuration for connection security

Attribute name	Description
IPSECRULE	IP security rules
TEXT	Text description

Table 25. Attributes that can be monitored for Network server descriptions for integrated network servers

Attribute name	Description
CFGFILE	Configuration file
CODEPAGE	ASCII code page representing the character set to be used by this network server
EVTLOG	Event log
MSGQ	Message queue
NWSSTGL	Storage space links
PRPDMNUSR	Propagate domain user
RSRCNAME	Resource name
RSTDDEVRSC	Restricted device resources
SHUTDTIMO	Shut down time out
SYNCTIME	Synchronize date and time
TCPDMNNAME	TCP/IP local domain name
TCPHOSTNAM	TCP/IP host name
TCPPORTCFG	TCP/IP port configuration
TCPNAMSVR	TCP/IP name server system
TEXT	Text description
VRYWAIT	Vary on wait
WINDOWSNT	Windows network server description

Table 26. Attributes that can be monitored for network server descriptions for iSCSI connections

Attribute name	Description
ACTTMR	Activation timer
CFGFILE	Configuration file

Table 26. Attributes that can be monitored for network server descriptions for iSCSI connections (continued)

Attribute name	Description
CMNMSGQ	Communications message queue
CODEPAGE	ASCII code page representing the character set to be used by this network server
DFTSECRULE	Default IP security rule
DFTSTGPTH	Default storage path
EVTLOG	Event log
MLTPTHGRP	Multi-path group
MSGQ	Message queue
NWSCFG	Network server configuration
NWSSTGL	Storage space links
PRPDMNUSR	Propagate domain user
RMVMEDPTH	Removable media path
RSRCNAME	Resource name
RSTDDEVRSC	Restricted device resources
SHUTDTIMO	Shut down time out
STGPTH	iSCSI storage paths of the network server
SVROPT	Serviceability options
SYNCTIME	Synchronize date and time
TCPDMNNAME	TCP/IP local domain name
TCPHOSTNAM	TCP/IP host name
TCPNAMSVR	TCP/IP name server system
TCPPORTCFG	TCP/IP port configuration
TEXT	Text description
VRTETHCTLP	Virtual Ethernet control port
VRTETHPTH	Virtual Ethernet path
VRYWAIT	Vary on wait

Table 27. Attributes that can be monitored for network server storage spaces

Attribute name	Description
SIZE	Size
TEXT	Text description
TOTALFILES	Total files

Table 28. Attributes that can be monitored for network server host adapter device descriptions

Attribute name	Description
CMNRCYLMT	Recovery limits
LCLIFC	Associated local interface
MSGQ	Message queue
ONLINE	Online at IPL
RSRCNAME	Resource name

Table 28. Attributes that can be monitored for network server host adapter device descriptions (continued)

Attribute name	Description
TEXT	Text description

Table 29. Attributes that can be monitored for optical device descriptions

Attribute name	Description
MSGQ	Message queue
ONLINE	Online at IPL
RSRCNAME	Resource name
TEXT	Text description

*Table 30. Attributes that can be monitored for printer device descriptions for *LAN printers*

Attribute name	Description
ACTTMR	Activation timer
ADPTADR	LAN remote adapter address
ADPTTYPE	Adapter type
ADPTCNNTYP	Adapter connection type
AFP	Advanced function printing
CHRID	Character identifier
FONT	Font
FORMFEED	Formfeed
IMGCFG	Image configuration
INACTTMR	Inactivity timer
LNGTYPE	Language type
LOCADR	Location location address
MAXPNDRQS	Maximum pending request
MFRTYPMDL	Manufacturer type and model
MSGQ	Message queue
ONLINE	Online at IPL
PORT	Port number
PRTERRMSG	Print error message
PUBLISHINF	Publishing information
RMTLOCNAME	Remote location
SEPDRAWER	Separator drawer
SEPPGM	Separator program
SWTLINLST	Switched line list
SYSDRVPGM	System driver program
TEXT	Text description
TRANSFORM	Host printer transform
USRDFNOBJ	User-defined object
USRDFNOPT	User-defined options
USRDRVPGM	User-defined driver program

Table 30. Attributes that can be monitored for printer device descriptions for *LAN printers (continued)

Attribute name	Description
USRDTATFM	Data transform program
WSCST	Workstation customizing object

Table 31. Attributes that can be monitored for printer device descriptions for *VRT printers

Attribute name	Description
CHRID	Character identifier
FORMFEED	Form feed
IGCFEAT	DBCS FEATURE
IMGCFG	Image configuration
MAXLENRU	Maximum length of request unit
MFRTYPMDL	Manufacturer type and model
MSGQ	Message queue
ONLINE	Online at IPL
PRTERRMSG	Print error message
PUBLISHINF	Publishing information
SEPDRAWER	Separator drawer
SEPPGM	Separator program
TEXT	Text description
TRANSFORM	Host print transform
USRDFNOBJ	User-defined object
USRDFNOPT	User-defined options
USRDRVPGM	User-defined driver program
USRDTAFM	Data transform program
WSCST	Workstation customizing object
SEPPGM	Separator program
SWTLINLST	Switched line list
SYSDRVPGM	System driver program
TEXT	Text description
TRANSFORM	Host printer transform
USRDFNOBJ	User-defined object
USRDFNOPT	User-defined options
USRDRVPGM	User-defined driver program
USRDTATFM	Data transform program
WSCST	Workstation customizing object

Table 32. Attributes that can be monitored for subsystem descriptions

Attribute name	Description
AJE	Autostart job entry
CMNE	Online at IPL
JOBQE	Job queue

Table 32. Attributes that can be monitored for subsystem descriptions (continued)

Attribute name	Description
MAXJOBS	Maximum number of jobs
PJE	Prestart job entry
RMTLOCNAME	Remote location name
RTGE	Routing entry
SGNDSPF	Sign on display
SYSLIBLE	Subsystem library
TEXT	Text description
WSNE	Workstation name entry
WSTE	Workstation type entry

Table 33. Attributes that can be monitored for system environment variables

Any *SYS level environment variable can be monitored. The attribute and resource name are both the same as the environment variable's name.
Note: Each environment variable is treated as its own monitored resource entry. For these, the resource type and attribute names are identical.

Table 34. Attributes that can be monitored for system values

Attribute name	Description
QACGLVL	Accounting level
QACTJOBITP	Allow jobs to be interrupted
QALWOBJRST	Prevents anyone from restoring a system-state object or an object that adopts authority
QALWUSRDMN	Allows user domain objects
QASTLVL	Assistance level
QATNPGM	Attention program
QAUDCTL	Audit control
QAUDENDACN	Audit journal error action
QAUDFRCLVL	Auditing force level
QAUDLVL	Auditing level
QAUDLVL2	Auditing level extension
QAUTOCFG	Automatic device configuration
QAUTORMT	Remote controllers and devices
QAUTOVRT	Automatic virtual device configuration
QCCSID	Coded character set identifier
QCFGMSGQ	Message queue for lines, controllers, and devices
QCHRID	Default graphic character set and code page used for displaying or printing data
QCHRIDCTL	Character identifier control for the job
QCMNRCYLMT	Automatic communications error recovery
QCNTRYID	Country or region identifier
QCRTAUT	Authority for new objects

Table 34. Attributes that can be monitored for system values (continued)

Attribute name	Description
QCRTOBJAUD	Auditing new objects
QCTLSBSD	Controlling subsystem or library
QCURSYM	Currency symbol
QDATFMT	Date format
QDATSEP	Date separator
QDBRCVYWT	Wait for database recovery before completing restart
QDECFMT	Decimal format
QDEVNAMING	Device naming convention
QDEVRCYACN	Device recovery action
QDSCJOBITV	Time out interval for disconnected jobs
QDSPSGNINF	Controls the display of sign-on information
QENDJOBLMT	Maximum time for immediate end
QFRCCVNRST	Force conversion on restore
QHSTLOGSIZ	History log file size
QIGCCDEFNT	Coded font name
QIGCFNTSIZ	Coded font point size
QINACTITV	Inactive job time-out interval
QINACTMSGQ	Timeout interval action
QIPLTYPE	Type of restart
QJOBMSGQFL	Job message queue full action
QJOBMSGQMX	Job message queue maximum size
QJOBMSGQSZ	Initial size of job message queue in kilobytes (KB)
QJOBMSGQTL	Maximum size of job message queue (in KB)
QJOBSPLA	Initial size of spooling control block for a job (in bytes)
QKBDBUF	Keyboard buffer
QKBDTYPE	Keyboard language character set
QLANGID	Default language identifier
QLIBLCKLVL	Lock libraries in a user job's library search list
QLMTDEVSSN	Limit device sessions
QLMTSECOFR	Limit security officer device access
QLOCALE	Locale
QLOGOUTPUT	Produce printer output for job log
QMAXACTLVL	Maximum activity level of the system
QMAXJOB	Maximum number of jobs that are allowed on the system
QMAXSGNACN	The system's response when the limit imposed by QMAXSIGN system value is reached
QMAXSIGN	Maximum number of not valid sign-on attempts allowed
QMAXSPLF	Maximum printer output files
QMLTTHDACN	When a function in a multithreaded job is not threadsafe
QPASTHRSVR	Available display station pass-through server jobs

Table 34. Attributes that can be monitored for system values (continued)

Attribute name	Description
QPRBFTR	Problem log filter
QPRBHLDITV	Minimum retention
QPRTDEV	Default printer
QPRTKEYFMT	Print key format
QPRTTXT	Up to 30 characters of text that can be printed at the bottom of listings and separator pages
QPWDCHGBLK	Minimum time between password changes
QPWDEXPITV	Number of days for which a password is valid
QPWDEXPWRN	Password expiration warning interval system
QPWDLMTACJ	Limits the use of adjacent numbers in a password
QPWDLMTCHR	Limits the use of certain characters in a password
QPWDLMTREP	Limits the use of repeating characters in a password
QPWDLVL	Password level
QPWDMAXLEN	Maximum number of characters in a password
QPWDMINLEN	Minimum number of characters in a password
QPWDPOSDIF	Controls the position of characters in a new password
QPWDRQDDGT	Require a number in a new password
QPWDRQDDIF	Controls whether the password must be different from the previous passwords
QPWDRULES	Password rules
QPWDVLDPGM	Password approval program
QPWRDWNLMT	Maximum time for immediate shutdown
QRCLSPLSTG	Automatically clean up unused printer output storage
QRETSVRSEC	Retain server security data indicator
QRMTSIGN	Remote sign-on
QRMTSRVATR	Remote service attribute
QSCANFS	Scan file systems
QSCANFSCTL	Scan control
QSCPFCONS	Console problem occurs
QSECURITY	System security level
QSETJOBATR	Set job attributes
QSFWERRLOG	Software error log
QSHRMEMCTL	Allow use of shared or mapped memory with write capability
QSPCENV	Default user environment
QSPLFACN	Spooled file action
QSRTSEQ	Sort sequence
QSRVDMP	Service log for unmonitored escape messages
QSSLCSL	Secure Sockets Layer cipher specification list
QSSLCSLCTL	Secure Sockets Layer cipher control
QSSLPCL	Secure Sockets Layer protocols

Table 34. Attributes that can be monitored for system values (continued)

Attribute name	Description
QSTRUPPGM	Set startup program
QSTSMSG	Display status messages
QSYSLIBL	System library list
QTIMSEP	Time separator
QTSEPOOL	Indicates whether interactive jobs should be moved to another main storage pool when they reach time slice end
Note: Each system value is treated as its own monitored resource entry. For these, the resource type and attribute names are identical.	

Table 35. Attributes that can be monitored for tape device descriptions

Attribute name	Description
ASSIGN	Assign device at vary on
MSGQ	Message queue
ONLINE	Online at IPL
RSRCNAME	Resource name
TEXT	Text description
UNLOAD	Unload device at vary off

Table 36. Attributes that can be monitored for token-ring descriptions

Attribute name	Description
ACTLANMGR	Activate LAN manager
ADPTADR	Local adapter address
AUTOCRTCTL	Autocreate controller
AUTODLTCTL	Autodelete controller
CMNRCYLMT	Recovery limits
COSTBYTE	Relative cost per byte for sending and receiving data on the line
COSTCNN	Relative cost of being connected on the line
DUPLEX	Duplex
ELYTKNRLS	Early token release
FCNADR	Functional address
LINESPEED	Line speed
LINKSPEED	Link speed
LOGCFGCHG	Log configuration changes
MAXCTL	Maximum controllers
MAXFRAME	Maximum frame size
MSGQ	Message queue
ONLINE	Online at IPL
PRPDLY	Propagation delay
RSRCNAME	Resource name

Table 36. Attributes that can be monitored for token-ring descriptions (continued)

Attribute name	Description
SECURITY	Security for line
SSAP	Source service access point (SSAP) information list
TRNINFBDN	Token-ring inform of beacon
TRNLOGLVL	TRLAN manager logging level
TRNMGRMODE	TRLAN manager mode
TEXT	Text description of the token-ring line
USRDFN1	First user-defined
USRDFN2	Second user-defined
USRDFN3	Third user-defined
VRYWAIT	Vary on wait

Table 37. Attributes that can be monitored for TCP/IP attributes

Attribute name	Description
ARPTIMO	Address resolution protocol (ARP) cache timeout
ECN	Enable explicit congestion notification (ECN)
IP6TMPAXP	IPv6 temporary address excluded prefix
IPDEADGATE	IP dead gateway detection
IPDTGFWD	IP datagram forwarding
IPPATHMTU	Path maximum transmission unit (MTU) discovery
IPQOSBCH	IP QoS datagram batching
IPQOSENB	IP QoS enablement
IPQOSTMR	IP QoS timer resolution
IPRSBTIMO	IP reassembly timeout
IPSRCRTG	IP source routing
IPTTL	IP time to live (hop limit)
LOGPCLERR	Log protocol errors
NFC	Network file cache
TCPCLOTIMO	TCP time-wait timeout
TCPCNNMSG	TCP close connection message
TCPKEEPALV	TCP keep alive
TCPMINRTM	TCP minimum retransmit time
TCPR1CNT	TCP R1 retransmission count
TCPR2CNT	TCP R2 retransmission count
TCPRCVBUF	TCP receive buffer size
TCPSNDBUF	TCP send buffer size
TCPURGPTR	TCP urgent pointer
UDPCKS	UDP checksum
Note: Each TCP/IP attribute is treated as its own monitored resource entry. For these, the resource type and attribute names are identical.	

Table 38. Attributes that can be monitored for user profiles

Attribute name	Description
ACGCDE	Accounting code
ASTLVL	Assistance level
ATNPGM	Attention program
CCSID	Coded character set ID
CHRIDCTL	Character identifier control
CNTRYID	Country or region ID
CURLIB	Current library
DLVRY	Delivery
DSPSGNINF	Display sign-on information
GID	Group ID number
GRPAUT	Group authority
GRPAUTTYP	Group authority type
GRPPRF	Group profile
HOMEDIR	Home directory
INLMNU	Initial menu
INLPGM	Initial program to call
JOBD	Job description
KBDBUF	Keyboard buffering
LANGID	Language ID
LCLPWDMGT	Local password management
LMTCPB	Limit capabilities
LMTDEVSSN	Limit device sessions
LOCALE	Locale
MAXSTG	Maximum allowed storage
MSGQ	Message queue
OUTQ	Output queue
OWNER	Owner
PASSWORD	User password
PRTDEV	Print device
PTYLMT	Highest schedule priority
PWDEXP	Set password to expired
PWDEXPITV	Password expiration interval
SETJOBATR	Locale job attributes
SEV	Severity code filter
SPCAUT	Special authority
SPCENV	Special environment
SRTSEQ	Sort sequence
STATUS	Status
SUPGRPPRF	Supplemental groups
TEXT	Text description

Table 38. Attributes that can be monitored for user profiles (continued)

Attribute name	Description
UID	User ID number
USRCLS	User class
USREXPDATE	User expiration date
USREXPITV	User expiration interval
USROPT	User options

Displaying monitored resource entry messages

Using the Cluster Resource Services graphical interface, you can display messages associated with monitored resource entries.

To display and view monitored resource entry messages, complete the following:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the Cluster Resource Services page, click **Work with Administrative Domains** to display a list of cluster administrative domains in the cluster.
5. On the Administrative Domains page, click the context icon next to the cluster administrative domain name, and select **Monitored Resource Entries**.

   Note: The **Monitored Resource Entries** action is available only if the node that you are managing is part of the cluster administrative domain. The current list of monitored resource types is shown.
6. In the list of monitored resource types, click the context icon next to the name, and select **Attributes**. The MRE object list is shown.
7. Click the context icon next to the MRE object, such as a user profile or system value, and select **Display Values**.

Chapter 9. Managing switched disks

Switched disks are independent disk pools that have been configured as part of a device cluster resource group (CRG). Ownership of data and applications that stored in a switched disk can be switched to other systems that have been defined in the device CRG. Switched disk technology provides high availability during planned and some unplanned outages.

Making a disk pool unavailable

You can select an independent disk pool to make it unavailable (vary it off). You cannot access any of the disk units or objects in the independent disk pool or its corresponding database until it is made available (varied on) again. The pool can be made available again on the same system or another system in the recovery domain of the cluster resource group.

Important: Before an independent disk pool can be made unavailable, no jobs can hold reservations on the disk pool. See Release job reservations on an independent disk for details on determining whether jobs are using an independent disk pool and how to release the job reservations.

When making a UDFS disk pool unavailable using System i Navigator, messages might be generated that require a response in the character-based interface. System i Navigator will not provide any indication that a message is waiting.

To make an independent disk pool unavailable:

1. In System i Navigator, expand **My Connections** (or your active environment).
2. Expand the System i you want to examine,**Configuration and Service** > **Hardware** > **Disk Units**.
3. Expand **Disk Pools**.
4. Right-click the disk pool you want to make unavailable and select **Make Unavailable**.
5. From the dialog box that displays, click **Make Unavailable** to make the disk pool unavailable.

You can use the Vary Configuration (VRYCFG) command in the character-based interface to make the disk pool unavailable.

Use the Display ASP Status (DSPASPSTS) command to identify where a step is in the process.

Use the Control ASP Access (QYASPCTLAA) API to restrict the processes that have access to the ASP.

Use the Start DASD Management Operation (QYASSDMO) API to reduce the amount of time it takes to make a disk pool unavailable.

Making your hardware switchable

In an i5/OS high-availability environment, you must make an external expansion unit switchable.

When you are using independent disk pools in a switchable environment, the associated hardware must be authorized to switch as well. Depending on your environment, this can include frame or units or IOPs and their associated resources. Refer to the following steps that apply to your switchable environment.

Making frame or unit switchable

An independent disk pool can contain disk units within several expansion units. If you have a stand-alone expansion unit that contains disk units included in an independent disk pool, you must

authorize the expansion unit to grant access to other systems. This is called making an expansion unit switchable. If you do not want other systems to be able to access the stand-alone expansion unit, you must make the expansion unit private.

To make a frame or unit switchable, follow these steps:

1. In System i Navigator, expand **My Connections** (or your active environment).
2. Expand the system you want to examine, **Configuration and Service** > **Hardware** > **Disk Units** > **By Location** and select the frame or disk unit that you want to make switchable.
3. Right-click a highlighted frame or disk unit and select **Make Switchable**.
4. Follow the instructions on the dialog box that is displayed.

Making IOP switchable

To allow an IOP to be switched, the bus containing the IOP that controls the disk units to be switched must be owned by the primary node (owned shared). The backup node must also use the bus (use bus shared). See Dynamically switching IOPs between partitions for more information.

To complete this task, you need a Service Tools user profile with administration authority to the System Partitions function in dedicated service tools (DST). For more information about obtaining logical partition privileges, refer to Logical partition authority.

To change the ownership type for a bus by using Management Central, follow these steps:

1. In System i Navigator, expand **My Connections** (or your active environment).
2. Select the primary partition of the system.
3. Expand **Configuration and Service** and select **Logical Partitions**.
4. Right-click the **Logical Partition** and select **Configure Partitions**.
5. In the Configure Logical Partitions window, right-click the bus for which you want to change ownership and select **Properties**.
6. Select the **Partitions** page.
7. Select the partition that owns the bus in **Owning logical partition**, and then select the ownership type in **Sharing**. If the ownership type is shared, the partitions that share the bus appear in the list. Click Help if you need more information about these options.
8. Click **OK**.

Making I/O pool switchable with Hardware Management Console

If you are using the Hardware Management Console to manage your logical partitions, you must create an I/O pool that includes the IOP, IOA, and all attached resources to allow an independent disk pool to be switchable between partitions. You must grant access to each partition that you want to own the independent disk pool by assigning the I/O pool in each partition profile.

To create an I/O pool that can be switched between partitions, follow these steps:

1. Open the Logical Partition Profile Properties window to change partition profile properties and assign resources to an I/O pool.
2. Click the **Physical I/O** tab.
3. In the Profile I/O devices column, expand the bus that contains the IOP that you want to make switchable.
4. Select the IOP that you want to assign to an I/O pool. The IOP must be *desired* (no check mark in the **Required** column).
5. Click the I/O pool column so that the cursor appears in the row of the IOP you want to assign to an I/O Pool, and type the number for the I/O pool.

6. Repeat these steps to add each IOA and resource under the control of the IOP to the I/O pool.
7. Click **OK**.

Associating I/O pool with partitions

After you have added the resources to the I/O pool, complete the following steps to associate the I/O pool with each additional partition that you want to be able to own the independent disk pool in the switchable environment.

1. Open the Logical Partition Profile Properties window to change partition profile properties for each additional partition that needs to access the independent disk pool.
2. Click the **Physical I/O** tab.
3. Click **Advanced**.
4. In the I/O Pools window, in the **I/O pools to add** field, type the number of the I/O pool to which you assigned the resources that you want to switch with the independent disk pool.
5. Click **Add** > **OK**.

For the I/O pool changes to take effect, complete the following steps for each partition whose partition profile was changed:

1. Shut down the partition. See Restarting and shutting down IBM i in a logical partition.
2. Start the logical partition by activating the partition profile to reflect the changes.

Related concepts:

Dynamically switching IOPs between partitions

Logical partition authority

I/O pool

Related tasks:

Changing partition profile properties

Activating the partition profile

Restarting and shutting down i5/OS™ in a logical partition.

Quiescing an independent disk pool

In an i5/OS high-availability solution, independent disk pools are used to store resilient data and applications. Some system functions, such as performing backups, require that you temporarily suspend changes to that data while the operation occurs.

To decrease the amount of time it takes to quiesce an independent disk pool, you might want to hold batch job queues, end some subsystems, or send a break message to interactive users, advising them to postpone new work.

To quiesce an independent disk pool, complete these steps.

In a command line interface, enter the following command: **CHGASPACT ASPDEV(name) OPTION(*SUSPEND) SSPTIMO(30) SSPTIMOACN(*CONT),**, where *name* is the name of the independent disk pool that you want to suspend. In this command you are specifying to suspend the independent disk pool with a 30-second timeout, and to continue with the next step even if the timeout limit has been exceeded.

Resuming an independent disk pool

After you have quiesced an independent disk pool in an i5/OS high availability environment for backup operations, you will need to resume the independent disk pool to ensure changes made to the data during the quiesce are updated.

Complete these steps to resume an independent disk pool:

In a command line interface, enter the following command:**CHGASPACT ASPDEV(name) OPTION(*RESUME),**, where `name` is the name of the independent disk pool that you want to resume.

Chapter 10. Managing cross-site mirroring

You can manage three cross-site mirroring technologies: geographic mirroring, metro mirror and global mirror. These cross-site mirroring technologies provide disaster recovery by copying critical data from disk units at the production site to disk units at a backup location.

Managing geographic mirroring

Use the following information to help you manage geographic mirroring. Geographic mirroring is a sub-function of cross-site mirroring, where data is mirrored to independent disk pools in an i5/OS environment.

Suspending geographic mirroring

If you need to end TCP communications for any reason, such as placing your system in restricted state, you should suspend geographic mirroring first. This action temporarily stops mirroring between systems in a high-availability solution.

When you suspend mirroring, any changes made on the production copy of the independent disk pool are not being transmitted to the mirror copy.

Note: When you resume geographic mirroring, synchronization is required between the production and mirror copies. If geographic mirroring was suspended without tracking, then full synchronization occurs. This can be a lengthy process.

Suspending geographic mirroring when IBM PowerHA for i is installed

To suspend geographic mirroring with IBM Systems Director Navigator for i, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i window.
4. Select **Disk Pools**.
5. Select the production copy of the **Disk Pool** that you want to suspend.
6. From the **Select Actions** menu, select **Sessions**.
7. Select the session that you want to suspend.
8. From the **Select Actions** menu, select **Suspend with tracking** or **Suspend without tracking**.

Suspending geographic mirroring when IBM PowerHA for i is not installed

To suspend geographic mirroring with System i Navigator, follow these steps:

1. In System i Navigator, expand **My Connections** (or your active environment).
2. Expand the system that owns the production copy of the geographically mirrored disk pool that you want to suspend.
3. Expand **Configuration and Service** > **Hardware** > **Disk Units** > **Disk Pools**.
4. Right-click the production copy of the **Disk Pool** you want to suspend and select **Geographic Mirroring** > **Suspend Geographic Mirroring**.

If you suspend with tracking, the system attempts to track changes made to those disk pools. This might shorten the length of the synchronization process by performing partial synchronization when you resume geographic mirroring. If tracking space is exhausted, then when you resume geographic mirroring, complete synchronization is required.

Note: If you suspend geographic mirroring without tracking changes, then when you resume geographic mirroring, a complete synchronization is required between the production and mirror copies. If you suspend geographic mirroring and you do track changes, then only a partial synchronization is required. Complete synchronization can be a lengthy process, anywhere from one to several hours or longer. The length of time it takes to synchronize is dependent on the amount of data being synchronized, the speed of TCP/IP connections, and the number of communication lines used for geographic mirroring.

Resuming geographic mirroring

If you suspend geographic mirroring, you must resume it in order to reactivate mirroring between the production and mirrored copies again.

Note: When you resume geographic mirroring, the production and mirror copies are synchronized concurrent with performing geographic mirroring. Synchronization can be a lengthy process. If a disk pool becoming unavailable interrupts synchronization, then synchronization continues from where it was interrupted when the disk pool becomes available again. When an interrupted synchronization is continued, the first message (CPI0985D) states that the synchronization is 0% complete.

Resuming geographic mirroring when IBM PowerHA for i is installed

To resume geographic mirroring with IBM Systems Director Navigator for i, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i window.
4. Select **Disk Pools**.
5. Select the production copy of the **Disk Pool** that you want to resume.
6. From the **Select Actions** menu, select **Sessions**.
7. Select the session that you want to resume.
8. From the **Select Actions** menu, select **Resume**.

Resuming geographic mirroring when IBM PowerHA for i is not installed

To resume geographic mirroring using System i Navigator, follow these steps:

1. In System i Navigator, expand **My Connections** (or your active environment).
2. Expand the system that owns the production copy of the disk pool for which you want to resume geographic mirroring.
3. Expand **Configuration and Service** > **Hardware** > **Disk Units** > **Disk Pools**.
4. Right-click the **Disk Pool** you want to resume and select **Geographic Mirroring** > **Resume Geographic Mirroring**.

Detaching mirror copy

If you are using geographic mirroring and want to access the mirror copy to perform save operations or data mining, or to create reports, you must detach the mirror copy from the production copy.

You detach the mirror copy by accessing the production copy of the disk pool.

Detaching the mirror copy when IBM PowerHA for i is installed

To detach the mirror copy by using IBM Navigator for i, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
4. Select **Disk Pools**.
5. Select the production copy of the **Disk Pool** that you want to detach.
6. From the **Select Actions** menu, select **Sessions**.
7. Select the session that you want to detach.
8. From the **Select Actions** menu, select **Detach with tracking** or **Detach without tracking**.

Detaching the mirror copy when IBM PowerHA for i is not installed

It is recommended, but not required, that you make the independent disk pool unavailable to ensure the production copy is not altered while the detachment is being performed.

To detach the mirror copy by using System i Navigator, follow these steps:

1. In System i Navigator, expand **My Connections** (or your active environment).
2. Expand the system that owns the production copy of the disk pool from which you want to detach the mirror copy.
3. Expand **Configuration and Service** > **Hardware** > **Disk Units** > **Disk Pools**.
4. Right-click the production copy of the **Disk Pool** you want to detach and select **Geographic Mirroring** > **Detach Mirror Copy**.

If **Geographic Mirroring** > **Detach Mirror Copy** cannot be clicked because it is disabled, the mirror copy is not in sync with the production copy, Geographic mirroring must be resumed, the disk pool varied on, and production and mirror copies synchronized before the mirror copy can be detached.

Before you make the detached mirror copy available, you should create a second, unique device description for the independent disk pool that differentiates it from the production copy. A separate device description for the mirror copy prevents two instances of the same database in the network. It will also simplify work done outside of System i Navigator. Use the detached mirror copy device description to make the detached mirror copy available.

Reattaching mirror copy

If you detached the mirror copy and have completed your work with the detached mirror copy, you must reattach the detached mirror copy to resume using geographic mirroring.

You reattach the detached mirror copy by accessing the production copy of the disk pool. The detached mirror copy must be unavailable when you reattach it to the production copy.

Note: When you reattach the detached mirror copy, as of V6R1, there is the option to detach with tracking which only requires a partial synchronization on reattach.

Reattaching the mirror copy when IBM PowerHA for i is installed

To reattach the mirror copy with IBM Navigator for i, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Navigator for i window.

4. Select **Disk Pools**.
5. Select the production copy of the **Disk Pool** that you want to reattach.
6. From the **Select Actions** menu, select **Sessions**.
7. Select the session that you want to reattach.
8. From the **Select Actions** menu, select **Attach**.

Reattaching the mirror copy when IBM PowerHA for i is not installed

To reattach the mirror copy using System i Navigator, follow these steps:

1. In System i Navigator, expand **My Connections** (or your active environment).
2. Expand the system that owns the production copy of the disk pool to which you want to reattach the detached mirror copy.
3. Expand **Configuration and Service** > **Hardware** > **Disk Units** > **Disk Pools**.
4. Right-click the production copy of the **Disk Pool** you want to reattach and select **Geographic Mirroring** > **Reattach Mirror Copy**.

Deconfiguring geographic mirroring

If you no longer want the capability to use geographic mirroring for a specific disk pool or disk pool group, you can select to **Deconfigure Geographic Mirroring**. If you deconfigure geographic mirroring, the system stops geographic mirroring and deletes the mirror copy of the disk pools on the nodes in the mirror copy site.

The disk pool must be offline to deconfigure geographic mirroring.

Deconfigure geographic mirroring when IBM PowerHA for i is installed

To deconfigure geographic mirroring with IBM Systems Director Navigator for i, follow these steps:

1. In a Web browser, enter **http://mysystem:200**1, where **mysystem** is the host name of the system.
2. Log on to the system with your user profile and password.
3. Expand the system you want to examine, **Configuration and Service** > **Disk Pools**
4. End the ASP Session for your geographic mirroring configuration
   a. Click the arrow beside the disk pool you wish to deconfigure. Choose **Session** > **Open...**
   b. Select your ASP Session. Choose the Delete action. Press Go.
5. Deconfigure Geographic Mirroring for the ASP
   a. Click the arrow beside the disk pool you wish to deconfigure. Choose **Session** > **New** > **Geographic Mirroring** > **Deconfigure Geographic Mirroring**
   b. Click Deconfigure on the Confirmation screen
6. Update your cluster configuration, as follows:
   a. Remove the nodes associated with the mirror copy from the device cluster resource group (CRG) recovery domain.
   b. Remove the site name and data port IP addresses from the remaining nodes in the cluster.

Deconfigure geographic mirroring when IBM PowerHA for i is not installed

1. In System i Navigator, expand **My Connections** (or your active environment).
2. Expand the system you want to examine, **Configuration and Service** > **Hardware** > **Disk Units** > **Disk Pools**.
3. Right-click the production copy of the **Disk Pool** you want to deconfigure and select **Geographic Mirroring** > **Deconfigure Geographic Mirroring**.
4. Update your cluster configuration, as follows:

a. Remove the nodes associated with the mirror copy from the device cluster resource group (CRG) recovery domain.
b. Remove the site name and data port IP addresses from the remaining nodes in the cluster.

Related tasks:

"Removing nodes" on page 135
You might need to remove a node from a cluster if you are performing an upgrade of that node or if the node no longer needs to participate in the i5/OS high-availability environment.

Changing geographic mirroring properties

You can change information associated with geographic mirroring and edit the associated copy descriptions.

Changing geographic mirroring properties with IBM Systems Director Navigator for i5/OS

To edit the geographic mirroring session by using IBM Systems Director Navigator for i5/OS, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
4. Select **Disk Pools**.
5. Select the disk pool associated with the session.
6. From the **Select Actions** menu, select **Sessions**.
7. Select the session.
8. From the **Select Actions** menu, select **Properties**. To change an associated copy description, select the copy description and click **Edit**.

Changing geographic mirroring properties with System i Navigator

To change the geographic mirroring properties by using System i Navigator, follow these steps:

1. In System i Navigator, expand **My Connections** (or your active environment).
2. Expand the system that owns the production copy of the geographically mirrored disk pool associated with the geographic mirror session for which you want to edit the attributes,**Configuration and Service** > **Hardware** > **Disk Units** > **Disk Pools**.
3. Right-click the production copy of the **Disk Pool** for which you want to edit the attributes and select **Sessions** > **Open**.
4. Right-click the production copy of the **Session** for which you want to edit the attributes and select **Properties**. To change an associated copy description, select the copy description and click **Edit**.

Managing metro mirror sessions

In i5/OS high availability environment that use IBM System Storage metro mirror technology, you must configure a metro mirroring session between the i5/OS systems and the external disk units with metro mirror configured. From the system, you can manage these sessions.

Suspending metro mirror sessions

You might need to suspend metro mirror sessions to perform maintenance on the system.

To suspend a metro mirror session, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.

3. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
4. Select **Disk Pools**.
5. Select the disk pool that you want to suspend.
6. From the **Select Actions** menu, select **Sessions**.
7. Select the session that you want to suspend.
8. From the **Select Actions** menu, select **Suspend**.

Resuming Metro Mirror sessions

After you have completed routine operations, such as performing maintenance on your system, you need to resume a suspended Metro Mirror session to re-enable high availability.

To resume a suspended metro mirroring session, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
4. Select **Disk Pools**.
5. Select the disk pool that is suspended.
6. From the **Select Actions** menu, select **Sessions**.
7. Select the session that is suspended.
8. From the **Select Actions** menu, select **Resume**.

Deleting metro mirror session

You can delete the metro mirror session to no longer use the session for high availability and disaster recovery.

To delete a metro mirror session, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
4. Select **Disk Pools**.
5. Select the disk pool associated with the session that you want to delete.
6. From the **Select Actions** menu, select **Sessions**.
7. Select the session that you want to delete.
8. From the **Select Actions** menu, select **Delete**.

Displaying or changing Metro Mirror properties

Display information about a metro mirroring session to change the associated copy descriptions.

To change the metro mirroring properties with IBM Navigator for i, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i window.
4. Select **Disk Pools**.
5. Select the disk pool associated with the session.
6. From the **Select Actions** menu, select **Sessions**.
7. Select the session.

8. From the **Select Actions** menu, select **Properties**. To change an associated copy description, select the copy description and click **Edit**.

Managing global mirror

In i5/OS high availability environment that use IBM System Storage global mirror technology, you must configure a global mirroring session between the i5/OS systems and the external disk units with global mirror configured. From the system, you can manage these sessions.

Suspending global mirror sessions

You might need to suspend global mirror sessions to perform maintenance on the system.

To suspend a global mirroring session, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
4. Select **Disk Pools**.
5. Select the disk pool that you want to suspend.
6. From the **Select Actions** menu, select **Sessions**.
7. Select the session that you want to suspend.
8. From the **Select Actions** menu, select **Suspend**.

Resuming Global Mirror sessions

After you have completed routine operations, such as performing maintenance on your system, you need to resume a suspended Global Mirror session to re-enable high availability.

To resume a suspended global mirroring session, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
4. Select **Disk Pools**.
5. Select the disk pool that is suspended.
6. From the **Select Actions** menu, select **Sessions**.
7. Select the session that is suspended.
8. From the **Select Actions** menu, select **Resume**.

Deleting global mirror sessions

You can delete the global mirror session to no longer use the session for high availability and disaster recovery.

To delete a global mirroring session, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
4. Select **Disk Pools**.
5. Select the disk pool associated with the session that you want to delete.
6. From the **Select Actions** menu, select **Sessions**.
7. Select the session that you want to delete.

8. From the **Select Actions** menu, select **Delete**.

Changing Global Mirror session properties

Display information about a Global Mirror session to change the associated copy descriptions.

To change the Global Mirror properties with IBM Navigator for i, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Navigator for i window.
4. Select **Disk Pools**.
5. Select the disk pool associated with the session.
6. From the **Select Actions** menu, select **Sessions**.
7. Select the session.
8. From the **Select Actions** menu, select **Properties**. To change an associated copy description, select the copy description and click **Edit**.

Managing switched logical units (LUNs)

Switched logical units are independent disk pools created from logical units created in an IBM System Storage DS8000 or DS6000 that have been configured as part of a device cluster resource group (CRG).

Ownership of data and applications that stored in a switched logical unit can be switched to other systems that have been defined in the device CRG. Switched disk technology provides high availability during planned and some unplanned outages.

Making switched logical units (LUNs) available and unavailable

You can select an independent disk pool to make it unavailable or available. You cannot access any of the disk units or objects in the independent disk pool or its corresponding database until it is made available again. The pool can be made available again on the same system or another system in the recovery domain of the cluster resource group.

An independent disk pool can be made unavailable by varying it off. Access to any of the disk units or objects in the independent disk pool or its corresponding database are not available, until they are varied on. The pool can be made available on the same system or another system in the recovery domain of the cluster resource group.

Quiescing an independent disk pool

In an i5/OS high-availability solution, independent disk pools are used to store resilient data and applications. Some system functions, such as performing backups, require that you temporarily suspend changes to that data while the operation occurs.

To decrease the amount of time it takes to quiesce an independent disk pool, you might want to hold batch job queues, end some subsystems, or send a break message to interactive users, advising them to postpone new work.

To quiesce an independent disk pool, complete these steps.

In a command line interface, enter the following command: **CHGASPACT ASPDEV(name) OPTION(*SUSPEND) SSPTIMO(30) SSPTIMOACN(*CONT),**, where *name* is the name of the independent disk pool that you want to suspend. In this command you are specifying to suspend the independent disk pool with a 30-second timeout, and to continue with the next step even if the timeout limit has been exceeded.

Resuming an independent disk pool

After you have quiesced an independent disk pool in an i5/OS high availability environment for backup operations, you will need to resume the independent disk pool to ensure changes made to the data during the quiesce are updated.

Complete these steps to resume an independent disk pool:

In a command line interface, enter the following command:**CHGASPACT ASPDEV(name) OPTION(*RESUME),** where `name` is the name of the independent disk pool that you want to resume.

Chapter 11. Managing the FlashCopy technology

FlashCopy is a IBM System Storage technology that allows you to take a point-in-time copy of external disk units. In i5/OS high availability solutions which use metro or global mirror, The FlashCopy technology can be used for backup window reduction by taking a copy of data which then can be backed up to media. To use the FlashCopy technology, a session must be created between the system and the external storage units.

Configuring a FlashCopy session

For i5/OS high-availability environments that use IBM System Storage technology, you can configure a FlashCopy session to create a point-in-time copy of data.

For information on using the FlashCopy feature on IBM System Storage DS8000, see IBM System Storage DS8000 Information Center.

To configure a FlashCopy session, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
4. Select **Disk Pools**.
5. Select the disk pool that you want to use as the source copy.
6. From the **Select Actions** menu, select **New Session**.
7. Follow the wizard's instructions to complete the task.

Updating a FlashCopy session

You can update a FlashCopy session when you are performing resynchronization of FlashCopy volumes on your IBM System Storage external storage units. Resynchronization allows you to make a copy without recopying the entire volume. This process is only possible with a persistent relationship, whereby the storage unit continually tracks updates to the source and target volumes. With persistent relationships, the relationship between the source and target volumes is maintained after the background copy has completed. The FlashCopy session created on the i5/OS provides a means to manage and monitor activity related to the FlashCopy session on the IBM System Storage units.

To update a FlashCopy session, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
4. Select **Disk Pools**.
5. Select the disk pool associated with the session that you want to update.
6. From the **Select Actions** menu, select **Sessions**.
7. Select the session that you want to update.
8. From the **Select Actions** menu, select **Update FlashCopy**.

Reattaching a FlashCopy session

Reattach a FlashCopy session.

To reattach a FlashCopy session, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
4. Select **Disk Pools**.
5. Select the disk pool associated with the session that you want to reattach.
6. From the **Select Actions** menu, select **Sessions**.
7. Select the session that you want to reattach.
8. From the **Select Actions** menu, select **Reattach FlashCopy**.

Detaching a FlashCopy session

You can detach the target volumes from the source for a selected FlashCopy session.

To detach target volumes from the source for a selected a FlashCopy session, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
4. Select **Disk Pools**.
5. Select the disk pool associated with the session that you want to detach.
6. From the **Select Actions** menu, select **Sessions** .
7. Select the session from which you want to detach target and source volumes.
8. From the **Select Actions** menu, select **Detach FlashCopy** .

Deleting a FlashCopy session

Delete a FlashCopy session.

To delete a FlashCopy session, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
4. Select **Disk Pools**.
5. Select the disk pool associated with the session that you want to delete.
6. From the **Select Actions** menu, select **Sessions**.
7. Select the session that you want to delete.
8. From the **Select Actions** menu, select **Delete**.

Restoring data from a FlashCopy session

After a FlashCopy session has been completed on the IBM System Storage units, you can restore that data from target volume to the source volume in the event of an outage at the source copy of data. To do this you need to reverse the FlashCopy session that is created on i5/OS. However, reversing the session copies data from the target back to the source and returns the source to it an earlier version.

Attention: Reversing a FlashCopy session backs out the changes made on the source copy by copying the target's data back to the source. This returns the source to that earlier point in time.

To reverse a FlashCopy session, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
4. Select **Disk Pools**.
5. Select the disk pool of the source copy.
6. From the **Select Actions** menu, select **Open Sessions**.
7. Select the session.
8. From the **Select Actions** menu, select **Reverse FlashCopy**.

Changing FlashCopy properties

Display information about a FlashCopy session to change the associated copy descriptions.

To change information about a FlashCopy session, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Configuration and Service** from your IBM Systems Director Navigator for i5/OS window.
4. Select **Disk Pools**.
5. Select the disk pool associated with the session
6. From the **Select Actions** menu, select **Sessions**.
7. Select the session.
8. From the **Select Actions** menu, select **Properties**. To change an associated copy description, select the copy description and click **Edit**.

Chapter 12. Troubleshooting your high availability solution

After you have configured you i5/OS high-availability solution, you may encounter problems with different technologies, including clusters and cross-site mirroring.

Troubleshooting clusters

Find error recovery solutions for problems that are specific to clusters.

At times, it may appear that the cluster is not working properly. This topic covers information about problems that you may encounter with clusters.

Determine if a cluster problem exists

Start here to diagnose your cluster problems.

At times, it may seem that your cluster is not operating correctly. When you think a problem exists, you can use the following to help determine if a problem exists and the nature of the problem.

- **Determine if clustering is active on your system.**

  To determine if cluster resource services is active, look for the two jobs - QCSTCTL and QCSTCRGM - in the list of system jobs. If these jobs are active, then cluster resource services is active. You can use the Work Management function in IBM Director Navigator for i5/OS or in System i Navigator to View jobs or use the **WRKACTJOB (Work with Active Jobs)** command to do this. You can also use the **DSPCLUINF (Display Cluster Information)** command to view status information for the cluster.

  – Additional jobs for cluster resource services may also be active. Cluster jobs provides information about how cluster resource services jobs are formatted.

- **Determine the cause of a CPFBB26 message.**

  ```
  Message . . . . : Cluster Resource Services not active or not responding.
  Cause . . . . . : Cluster Resource Services is either not active or cannot
  respond to this request because a resource is unavailable or damaged.
  ```

  This error can mean that either the CRG job is not active or the cluster is not active. Use the **DSPCLUINF** (Display Cluster Information) command to determine if the node is active. If the node is not active, start the cluster node. If it is active, you should also check the CRG to determine whether the CRG has problems.

  Look for the CRG job in the list of system jobs. You can use the Work Management function in IBM Director Navigator for i5/OS or in System i Navigator to View jobs or use the **WRKACTJOB (Work with Active Jobs)** command to do this. You can also use the **DSPCRGINF (Display CRG Information**) command to view status information for the specific CRG, by specifying the CRG name in the command. If the CRG job is not active, look for the CRG job log to determine the cause of why it was ended. Once the problem is fixed, you could restart the CRG job with **CHGCLURCY (Change Cluster Recovery) command** or by ending and restarting cluster on that node.

- **Look for messages indicating a problem.**

  – Ensure that you can review all messages associated with a cluster command, by selecting F10 which toggles between "Include detailed messages" and "Exclude detailed messages". Select to include all detailed messages and review them to determine if other actions are necessary.

  – Look for inquiry messages in QSYSOPR that are waiting for a response.

  – Look for error messages in QSYSOPR that indicate a cluster problem. Generally, these will be in the CPFBB00 to CPFBBFF range.

  – Display the history log (**DSPLOG** CL command) for messages that indicate a cluster problem. Generally, these will be in the CPFBB00 to CPFBBFF range.

- **Look at job logs for the cluster jobs for severe errors.**

  These jobs are initially set with a logging level at (4 0 *SECLVL) so that you can see the necessary error messages. You should ensure that these jobs and the exit program jobs have the logging level set appropriately. If clustering is not active, you can still look for spool files for the cluster jobs and exit program jobs.
- **If you suspect some kind of hang condition, look at call stacks of cluster jobs.**

  Determine if there is any program in some kind of DEQW (dequeue wait). If so, check the call stack of each thread and see if any of them have getSpecialMsg in the call stack.
- **Check for cluster vertical licensed internal code (VLIC) logs entries.**

  These log entries have a 4800 major code.
- **Use NETSTAT command to determine if there are any abnormalities in your communications environment.**

  NETSTAT returns information about the status of Internet Protocol network routes, interfaces, TCP connections, and UDP ports on your system.
  - Use Netstat Option 1 (Work with TCP/IP interface status) to ensure that the IP addresses chosen to be used for clustering show an 'Active' status. Also ensure that the LOOPBACK address (127.0.0.1) is also active.
  - Use Netstat Option 3 (Work with TCP/IP Connection Status) to display the port numbers (F14). Local port 5550 should be in a 'Listen' state. This port must be opened using the STRTCPSVR *INETD command evidenced by the existence of a QTOGINTD (User QTCP) job in the Active Jobs list. If clustering is started on a node, local port 5551 must be opened and be in a '*UDP' state. If clustering is not started, port 5551 must not be opened or it will, in fact, prevent the successful start of clustering on the subject node.
- Use PING to verify if there is a communications problem. If you try to start a cluster node and there is a communications problem, you may receive an internal clustering error (CPFBB46). However, PING does not work between IPv4 and IPv6 addresses, or if a firewall is blocking it.

Gathering recovery information for a cluster

You can use the **Work with Cluster (WRKCLU)** command to collect information for a complete picture of your cluster. This information can be used to aid in error resolution.

The **Work with Cluster (WRKCLU)** command is used to display and to work with cluster nodes and objects. When you run this command, the Work with Cluster display is shown. In addition to displaying nodes in a cluster and cluster information, you can use this command to view cluster information and to gather data about your cluster

To gather error recovery information, complete these steps:

1. On a character-based interface, type `WRKCLU OPTION(OPTION)`. You can specify the following options to indicate with which cluster status information you want to work with.

   ***SELECT**
   Display the Work with Cluster menu.

   ***CLUINF**
   Display cluster information.

   ***CFG** Display the performance and configuration parameters for the cluster.

   ***NODE**
   Display the Work with Cluster Nodes panel which is a list of nodes in the cluster.

   ***DEVDMN**
   Display the Work with Device Domains panel which is a list of device domains in the cluster.

   ***CRG** Display the Work with Cluster Resource Groups panel which is a list of cluster resource groups in the cluster.

***ADMDMN**
Display the Work with Administrative Domains panel which is a list of administrative domains in the cluster.

***SERVICE**
Gathers related trace and debug information for all cluster resource service jobs in the cluster. This information is written to a file with a member for each cluster resource service job. Use this option only when directed by your service provider. It will display a prompt panel for the **Dump Cluster Trace (DMPCLUTRC)**.

Common cluster problems

Lists some of the most common problems that can occur in a cluster, as well as ways to avoid and recover from them.

The following common problems are easily avoidable or easily correctable.

You cannot start or restart a cluster node

This situation is typically due to some problem with your communications environment. To avoid this situation, ensure that your network attributes are set correctly, including the loopback address, INETD settings, ALWADDCLU attibute, and the IP addresses for cluster communications.

- The ALWADDCLU network attribute must be appropriately set on the target node if trying to start a remote node. This should be set to either *ANY or *RQSAUT depending on your environment.
- The IP addresses chosen to be used for clustering locally and on the target node must show an *Active* status.
- The LOOPBACK address (127.0.0.1) locally and on the target node must also be active.
- Verify that network routing is active by attempting to PING using the IP addresses used for clustering on the local and remote nodes; however, PING does not work between IPv4 and IPv6 addresses, or if a firewall is blocking it. If any cluster node uses an IPv4 address, than every node in the cluster needs to have an active IPv4 address (not necessarily configured as a Cluster IP address) that can route to and send TCP packets to that address. Also, if any cluster node uses an IPv6 address, than every node in the cluster needs to have an active IPv6 address (not necessarily configured as a Cluster IP address) that can route to and send TCP packets to that address.
- INETD must be active on the target node. When INETD is active, port 5550 on the target node should be in a *Listen* state. See INETD server for information about starting the INETD server.
- Prior to attempting to start a node, port 5551 on the node to be started must not be opened or it will, in fact, prevent the successful start of clustering on the subject node.

You end up with several, disjointed one-node clusters

This can occur when the node being started cannot communicate with the rest of the cluster nodes. Check the communications paths.

The response from exit programs is slow.

A common cause for this situation is incorrect setting for the job description used by the exit program. The MAXACT parameter may be set too low so that, for example, only one instance of the exit program can be active at any point in time. It is recommended that this be set to *NOMAX.

Performance in general seems to be slow.

There are several common causes for this symptom.

- The most likely cause is heavy communications traffic over a shared communications line.

- Another likely cause is an inconsistency between the communications environment and the cluster message tuning parameters. You can use the Retrieve Cluster Resource Services Information (QcstRetrieveCRSInfo) API to view the current settings of the tuning parameters and the Change Cluster Resource Services (QcstChgClusterResourceServices) API to change the settings. Cluster performance may be degraded under default cluster tuning parameter settings if using old adapter hardware. The adapter hardware types included in the definition of *old* are 2617, 2618, 2619, 2626, and 2665. In this case, setting of the *Performance class* tuning parameter to *Normal* is desired.
- If all the nodes of a cluster are on a local LAN or have routing capabilities which can handle Maximum Transmission Unit (MTU) packet sizes of greater than 1,464 bytes throughout the network routes, large cluster message transfers (greater than 1,536K bytes) can be greatly speeded up by increasing the cluster tuning parameter value for *Message fragment size* to better match the route MTUs.

You cannot use any of the function of the new release.

If you attempt to use new release function and you see error message CPFBB70, then your current cluster version is still set at the prior version level. You must upgrade all cluster nodes to the new release level and then use the adjust cluster version interface to set the current cluster version to the new level. See Adjust the cluster version of a cluster for more information.

You cannot add a node to a device domain or access the System i Navigator cluster management interface.

To access the System i Navigator cluster management interface, or to use switchable devices, you must have IBM i Option 41, HA Switchable Resources installed on your system. You must also have a valid license key for this option.

You applied a cluster PTF and it does not seem to be working.

You should ensure that you have completed the following tasks after applying the PTF:

1. End the cluster
2. Signoff then signon

   The old program is still active in the activation group until the activation group is destroyed. All of the cluster code (even the cluster APIs) run in the default activation group.
3. Start the cluster

   Most cluster PTFs require clustering to be ended and restarted on the node to activate the PTF.

CEE0200 appears in the exit program joblog.

On this error message, the from module is QLEPM and the from procedure is Q_LE_leBdyPeilog. Any program that the exit program invokes must run in either *CALLER or a named activation group. You must change your exit program or the program in error to correct this condition.

CPD000D followed by CPF0001 appears in the cluster resource services joblog.

When you receive this error message, make sure the QMLTTHDACN system value is set to either 1 or 2.

Cluster appears hung.

Make sure cluster resource group exit programs are outstanding. To check the exit program, use the **WRKACTJOB (Work with Active Jobs)** command, then look in the Function column for the presence of PGM-QCSTCRGEXT.

Partition errors

Certain cluster conditions are easily corrected. If a cluster partition has occurred, you can learn how to recover. This topic also tells you how to avoid a cluster partition and gives you an example of how to merge partitions back together.

A cluster partition occurs in a cluster whenever contact is lost between one or more nodes in the cluster and a failure of the lost nodes cannot be confirmed. This is not to be confused with a partition in a logical partition (LPAR) environment.

If you receive error message CPFBB20 in either the history log (QHST) or the QCSTCTL joblog, a cluster partition has occurred and you need to know how to recover. The following example shows a cluster partition that involves a cluster made up of four nodes: A, B, C, and D. The example shows a loss of communication between cluster nodes B and C has occurred, which results in the cluster dividing into two cluster partitions. Before the cluster partition occurred, there were four cluster resource groups, which can be of any type, called CRG A, CRG B, CRG C, and CRG D. The example shows the recovery domain of each cluster resource group.

Table 39. Example of a recovery domain during a cluster partition

Node A	Node B	x	Node C	Node D
CRG A (backup1)	CRG A (primary)			
	CRG B (primary)		CRG B (backup1)	
	CRG C (primary)		CRG C (backup1)	CRG C (backup2)
CRG D (backup2)	CRG D (primary)		CRG D (backup1)	
Partition 1			**Partition 2**	

A cluster may partition if the maximum transmission unit (MTU) at any point in the communication path is less than the cluster communications tuneable parameter, message fragment size. MTU for a cluster IP address can be verified by using the **Work with TCP/IP Network Status (WRKTCPSTS)** command on the subject node. The MTU must also be verified at each step along the entire communication path. If the MTU is less than the message fragment size, either raise the MTU of the path or lower the message fragment size. You can use the Retrieve Cluster Resource Services Information (QcstRetrieveCRSInfo) API to view the current settings of the tuning parameters and the Change Cluster Resource Services (QcstChgClusterResourceServices) API to change the settings.

Once the cause of the cluster partition condition has been corrected, the cluster will detect the re-established communication link and issue the message CPFBB21 in either the history log (QHST) or the QCSTCTL joblog. This informs the operator that the cluster has recovered from the cluster partition. Be aware that once the cluster partition condition has been corrected, it may be a few minutes before the cluster merges back together.

Determining primary and secondary cluster partitions

In order to determine the types of cluster resource group actions that you can take within a cluster partition, you need to know whether the partition is a primary or a secondary cluster partition. When a partition is detected, each partition is designated as a primary or secondary partition for each cluster resource group defined in the cluster.

For primary-backup model, the primary partition contains the node that has the current node role of primary. All other partitions are secondary. The primary partition may not be the same for all cluster resource groups.

A peer model has the following partition rules:

- 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.
- If the cluster resource group is active and there are no peer nodes in the given partition, the cluster resource group will be ended in that partition.
- Operational changes are allowed in a secondary partition as long as the restrictions for the operational changes are met.
- No configuration changes are allowed in a secondary partition.

The restrictions for each Cluster Resource Group API are:

Table 40. Cluster Resource Group API Partition Restrictions

Cluster Resource Group API	**Allowed in primary partition**	**Allowed in secondary partitions**
Add Node to Recovery Domain	X	
Add CRG Device Entry		
Change Cluster Resource Group	X	
Change CRG Device Entry	X	X
Create Cluster Resource Group		
Delete Cluster Resource Group	X	X
Distribute Information	X	X
End Cluster Resource Group[1]	X	
Initiate Switchover	X	
List Cluster Resource Groups	X	X
List Cluster Resource Group Information	X	X
Remove Node from Recovery Domain	X	
Remove CRG Device Entry	X	
Start Cluster Resource Group[1]	X	
Note: 1. Allowed in all partitions for peer cluster resource groups, but only affects the partition running the API.		

By applying these restrictions, cluster resource groups can be synchronized when the cluster is no longer partitioned. As nodes rejoin the cluster from a partitioned status, the version of the cluster resource group in the primary partition is copied to nodes from a secondary partition.

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

When a partition is detected, the Add Cluster Node Entry, Adjust Cluster Version, and the Create Cluster API cannot be run in any of the partitions. The Add Device Domain Entry API can only be run if none of the nodes in the device domain are partitioned. All of the other Cluster Control APIs may be run in any partition. However, the action performed by the API takes affect only in the partition running the API.

Changing partitioned nodes to failed

Sometimes, a partitioned condition is reported when there really was a node outage. This can occur when cluster resource services loses communications with one or more nodes, but cannot detect if the nodes are still operational. When this condition occurs, a simple mechanism exists for you to indicate that the node has failed.

Attention: When you tell cluster resource services that a node has failed, it makes recovery from the partition state simpler. However, changing the node status to failed when, in fact, the node is still active and a true partition has occurred should not be done. Doing so can cause a node in more than one partition to assume the primary role for a cluster resource group. When two nodes think they are the primary node, data such as files or databases can become disjoint or corrupted if multiple nodes are each independently making changes to their copies of files. In addition, the two partitions cannot be merged back together when a node in each partition has been assigned the primary role.

When the status of a node is changed to Failed, the role of nodes in the recovery domain for each cluster resource group in the partition may be reordered. The node being set to Failed will be assigned as the last backup. If multiple nodes have failed and their status needs to be changed, the order in which the nodes are changed will affect the final order of the recovery domain's backup nodes. If the failed node was the primary node for a CRG, the first active backup will be reassigned as the new primary node.

When cluster resource services has lost communications with a node but cannot detect if the node is still operational, a cluster node will have a status of **Not communicating**. You may need to change the status of the node from **Not communicating** to **Failed**. You will then be able to restart the node.

To change the status of a node from **Not communicating** to **Failed**, follow these steps:

1. In a Web browser, enter `http://mysystem:2001`, where `mysystem` is the host name of the system.
2. Log on to the system with your user profile and password.
3. Select **Cluster Resource Services** from the IBM Systems Director Navigator for i window.
4. On the **Cluster Resource Services** page, select the **Work with Cluster Nodes** task to show a list of nodes in the cluster.
5. Click the **Select Action** menu and select **Change Status**. Change the status on the node to failed.

Related information:

Change Cluster Node (CHGCLUNODE) command

Change Cluster Node Entry (QcstChangeClusterNodeEntry) API

Partitioned cluster administrative domains

Consider the following information when working with partitioned cluster administrative domains.

If a cluster administrative domain is partitioned, changes continue to be synchronized among the active nodes in each partition. When the nodes are merged back together again, the cluster administrative domain propagates all changes made in every partition so that the resources are consistent within the active domain. There are several considerations regarding the merge processing for a cluster administrative domain:

- If all partitions were active and changes were made to the same resource in different partitions, the most recent change is applied to resource on all nodes during the merge. The most recent change is determined by using Coordinated Universal Time (UTC) from each node where a change initiated.
- If all partitions were inactive, the global values for each resource are resolved based on the last change made while any partition was active. The actual application of these changes to the monitored resources does not happen until the peer CRG that represents the cluster administrative domain is started.

- If some partitions were active and some were inactive prior to the merge, the global values representing changes made in the active partitions are propagated to the inactive partitions. The inactive partitions are then started, causing any pending changes made on the nodes in the inactive partitions to propagate to the merged domain.

Tips: Cluster partitions

Use these tips for cluster partitions.

1. The rules for restricting operations within a partition are designed to make merging the partitions feasible. Without these restrictions, reconstructing the cluster requires extensive work.
2. If the nodes in the primary partition have been destroyed, special processing may be necessary in a secondary partition. The most common scenario that causes this condition is the loss of the site that made up the primary partition. Use the example in recovering from partition errors and assume that Partition 1 was destroyed. In this case, the primary node for Cluster Resource Groups B, C, and D must be located in Partition 2. The simplest recovery is to use Change Cluster Node Entry to set both Node A and Node B to failed. See changing partitioned nodes to failed for more information about how to do this. Recovery can also be achieved manually. In order to do this, perform these operations:
   a. Remove Nodes A and B from the cluster in Partition 2. Partition 2 is now the cluster.
   b. Establish any logical replication environments needed in the new cluster. IE. Start Cluster Resource Group API/CL command, and so on.

   Since nodes have been removed from the cluster definition in Partition 2, an attempt to merge Partition 1 and Partition 2 will fail. In order to correct the mismatch in cluster definitions, run the Delete Cluster (QcstDeleteCluster) API on each node in Partition 1. Then add the nodes from Partition 1 to the cluster, and reestablish all the cluster resource group definitions, recovery domains, and logical replication. This requires a great deal of work and is also prone to errors. It is very important that you do this procedure only in a site loss situation.
3. Processing a start node operation is dependent on the status of the node that is being started:

   The node either failed or an End Node operation ended the node:
   a. Cluster resource services is started on the node that is being added
   b. Cluster definition is copied from an active node in the cluster to the node that is being started.
   c. Any cluster resource group that has the node being started in the recovery domain is copied from an active node in the cluster to the node being started. No cluster resource groups are copied from the node that is being started to an active node in the cluster.

   The node is a partitioned node:
   a. The cluster definition of an active node is compared to the cluster definition of the node that is being started. If the definitions are the same, the start will continue as a merge operation. If the definitions do not match, the merge will stop, and the user will need to intervene.
   b. If the merge continues, the node that is being started is set to an active status.
   c. Any cluster resource group that has the node being started in the recovery domain is copied from the primary partition of the cluster resource group to the secondary partition of the cluster resource group. Cluster resource groups may be copied from the node that is being started to nodes that are already active in the cluster.

Cluster recovery

Read about how to recover from other cluster failures that may occur.

Recovering from cluster job failures

Failure of a cluster resource services job is usually indicative of some other problem.

You should look for the job log associated with the failed job and look for messages that describe why it failed. Correct any error situations.

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

1. `CHGCLURCY CLUSTER(EXAMPLE)CRG(CRG1)NODE(NODE1)ACTION(*STRCRGJOB)` This command will cause cluster resource group job, CRG1, on node NODE1 to be submitted. To start the cluster resource group job on NODE1 requires clustering to be active on NODE1.
2. Restart clustering on the node.

If you are using a IBM Business Partner cluster management product, refer to the documentation that came with the product.

Related information:

Change Cluster Recovery (CHGCLURCY) command

Recovering a damaged cluster object

While it is unlikely you will ever experience a damaged object, it may be possible for cluster resource services objects to become damaged.

The system, if it is an active node, will attempt to recover from another active node in the cluster. The system will perform the following recovery steps:

For a damaged internal object

1. The node that has the damage ends.
2. If there is at least one other active node within the cluster, the damaged node will automatically restart itself and rejoin the cluster. The process of rejoining will correct the damaged situation.

For a damaged cluster resource group

1. The node that has a damaged CRG will fail any operation currently in process that is associated with that CRG. The system will then attempt to automatically recover the CRG from another active node.
2. If there is at least one active member in the recovery domain, the CRG recovery will work. Otherwise, the CRG job ends.

If the system cannot identify or reach any other active node, you will need to perform these recovery steps.

For a damaged internal object

You receive an internal clustering error (CPFBB46, CPFBB47, or CPFBB48).

1. End clustering for the node that contains the damage.
2. Restart clustering for the node that contains the damage. Do this from another active node in the cluster.
3. If Steps 1 and 2 do not solve the problem, remove the damaged node from the cluster.
4. Add the system back into the cluster and into the recovery domain for the appropriate cluster resource groups.

For a damaged cluster resource group

You receive an error stating that an object is damaged (CPF9804).

1. End clustering on the node that contains the damaged cluster resource group.
2. Delete the CRG by using the **DLTCRG** command.
3. If there is no other node active in the cluster that contains the CRG object, restore from media.
4. Start clustering on the node that contains the damaged cluster resource group. This can be done from any active node.

5. When you start clustering, the system resynchronizes all of the cluster resource groups. You may need to recreate the CRG if no other node in the cluster contains the CRG.

Recovering a cluster after a complete system loss

Use this information with the appropriate checklist in the Recovering your system topic for recovering your entire system after a complete system loss when your system loses power unexpectedly.

Scenario 1: Restoring to the same system

1. In order to prevent inconsistencies in the device domain information between the Licensed Internal Code and IBM i, it is recommended that you install the Licensed Internal Code by using option 3 (Install Licensed Internal Code and Recover Configuration).

   Note: For the Install Licensed Internal Code and Recover Configuration operation to succeed, you must have the same disk units -- with exception of the load source disk unit if it has failed. You must also be recovering the same release.
2. After you have installed the Licensed Internal Code, follow the Recovering Your Disk Configuration procedure in the *Recovering your system* topic. These steps will help you avoid having to reconfigure the disk pools.
3. After you have recovered your system information and are ready to start clustering on the node you just recovered, you must start clustering from the active node. This will propagate the most current configuration information to the recovered node.

Scenario 2: Restoring to a different system

After you have recovered your system information and checked the job log to make sure that all objects have restored, you must perform the following steps to obtain the correct cluster device domain configuration.

1. From the node you just restored, delete the cluster.
2. From the active node, perform these steps:
   a. Remove the recovered node from the cluster.
   b. Add the recovered node back into the cluster.
   c. Add the recovered node to the device domain.
   d. Create the cluster resource group or add the node to the recovery domain.

Recovering a cluster after a disaster

In the case of a disaster where all your nodes are lost, you will need to reconfigure your cluster.

In order to prepare for such a scenario, it is recommended that you save your cluster configuration information and keep a hardcopy printout of that information.

Restoring a cluster from backup tapes

During normal operations, you should never restore from a backup tape.

This is only necessary when a disaster occurs and all nodes were lost in your cluster. If a disaster should occur, you recover by following your normal recovery procedures that you have put in place after you created your backup and recovery strategy.

Troubleshooting cross-site mirroring

This information can help you solve problems related to cross-site mirroring that you might encounter.

Geographic mirroring messages

Review the geographic mirroring message descriptions and recoveries to resolve your geographic mirroring problems.

0x00010259

Description: Operation failed because the system did not find the mirror copy.

Recovery: Not all the nodes in the device domain responded. Make sure that clustering is active. If necessary, start clusters on the node. See "Starting nodes" on page 100 for details. Try the request again. If the problem persists, contact your technical support provider.

0x0001025A

Description: Not all of the disk pools in the disk pool group are geographically mirrored.

Recovery: If one disk pool in a disk pool group is geographically mirrored, all of the disk pools in the disk pool group must be geographically mirrored. Take one of the following actions:

1. Configure geographic mirroring for the disk pools which are not geographically mirrored.
2. Deconfigure geographic mirroring for the disk pools that are geographically mirrored.

0x00010265

Description: The detached mirrored copy is available.

Recovery: Make the detached mirrored copy unavailable and then try the reattach operation again.

0x00010380

Description: A disk unit is missing from the configuration of the mirror copy.

Recovery: Find or fix the missing disk unit in the mirror copy. Check the Product Activity Log on destination node. Reclaim IOP cache storage.

0x00011210

Description: The proposed secondary disk pool for the disk pool group is not geographically mirrored.

Recovery: If one disk pool in a disk pool group is geographically mirrored, all of the disk pools in the disk pool group must be geographically mirrored. You must configure geographic mirroring for the proposed secondary disk pool which is not geographically mirrored, either now or after completing this operation.

0x00011211

Description: Duplicate mirror copies exist.

Recovery: Check for locally mirrored disk units that may exist on two systems, Enterprise Storage Server® FlashCopy, or back level independent disk pool copies. See the Product Activity Log on the mirror copy node for more information. Eliminate duplication and try the request again. If the problem persists, contact your technical support provider, or see IBM i Technical Support for information about IBM support and services.

Installing IBM PowerHA for i licensed program

Before you can implement an IBM i high-availability solution, you must install the IBM PowerHA for i licensed program (5770-HAS) on each system that participates in high availability.

Before installing the IBM PowerHA for i licensed program, you should have completed the following installation requirements:

1. Install or upgrade to i 7.1 Operating System.
2. Install IBM i Operating System Option 41 (HA Switchable Resources).

To install the IBM PowerHA for i licensed program, complete the following steps:

1. Enter GO LICPGM from a command line.
2. At the Work with licensed programs display, select option 11 (Install licensed programs).
3. Select Product 5770-HAS, option *BASE to install IBM PowerHA for i Standard Edition. Press enter.
4. At the Install Options display, type in the name of your installation device as requested. Press enter to start the installation.
5. Using asynchronous geographic mirroring, metro mirroring, or global mirroring requires IBM PowerHA for i Enterprise Edition (option 1) be installed. Select Product 5770-HAS, option 1 to install IBM PowerHA for i Enterprise Edition. Press enter.

After successfully installing the IBM PowerHA for i licensed program, you need to restart the INETD server. For information about how to start INETD, see "Starting the INETD server" on page 90.

Code license and disclaimer information

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, IBM, ITS PROGRAM DEVELOPERS AND SUPPLIERS MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY.

UNDER NO CIRCUMSTANCES IS IBM, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

1. LOSS OF, OR DAMAGE TO, DATA;
2. DIRECT, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES; OR
3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

Appendix. 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.
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711

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, IBM License Agreement for Machine Code, 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 "Implementing high availability with the task-based approach" publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i5/OS.

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.

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

i5/OS
IBM
IBM (logo)
System i
System i5
System Storage
TotalStorage
FlashCopy

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.

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

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

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.

Printed in USA