IBM Systems - iSeries

Cluster APIs

Version 5 Release 4

IBM Systems - iSeries

Cluster APIs

Version 5 Release 4

Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 231.

Sixth Edition (February 2006)

This edition applies to version 5, release 4, modification 0 of IBM i5/OS (product number 5722-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.

© Copyright International Business Machines Corporation 1998, 2006. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

© Copyright IBM Corp. 1998, 2006

Cluster APIs

The Cluster APIs and exit program include:

- "Cluster Control APIs"
- "Cluster Resource Group APIs" on page 61
- "Clustered Hash Table APIs" on page 169
- "Cluster Resource Group Exit Program" on page 185

Other topics covered are:

- "Cluster APIs—Introduction" on page 218
- "Cluster Resource Services Characteristics" on page 219
- "Cluster Resource Services Job Structure" on page 220
- "Cluster APIs Use of User Queues" on page 221
- "Using Results Information" on page 223
- "Cluster Version" on page 224
- »«Cluster Administrative Domain

For additional information on clustering, see the Clusters topic.

See High Availability and Clusters for information on making applications highly available and cluster-proven applications.

APIs by category

APIs

These are the APIs for this category.

Cluster Control APIs

The cluster control APIs provide configuration, activation, and management functions for the cluster and nodes in the cluster.

For additional information, see:

- "Network Attributes" on page 3
- "Cluster Node Status" on page 3

The cluster control APIs are:

- "Add Cluster Node Entry (QcstAddClusterNodeEntry) API" on page 4 (QcstAddClusterNodeEntry) adds a node to the membership list of an existing cluster.
- "Add Device Domain Entry (QcstAddDeviceDomainEntry) API" on page 8 (QcstAddDeviceDomainEntry) adds a node entry to the membership list of a device domain.
- "Adjust Cluster Version (QcstAdjustClusterVersion) API" on page 11 (QcstAdjustClusterVersion) adjusts the current cluster version.
- "Change Cluster Node Entry (QcstChangeClusterNodeEntry) API" on page 14 (QcstChangeClusterNodeEntry) changes the fields in the cluster node entry.

© Copyright IBM Corp. 1998, 2006

- "Change Cluster Resource Services (QcstChgClusterResourceServices) API" on page 19 (QcstChgClusterResourceServices) tunes cluster performance and configuration parameters.
- "Create Cluster (QcstCreateCluster) API" on page 26 (QcstCreateCluster) creates a new cluster of one or more nodes.
- "Delete Cluster (QcstDeleteCluster) API" on page 31 (QcstDeleteCluster) deletes a cluster previously created by the Create Cluster API.
- "End Cluster Node (QcstEndClusterNode) API" on page 33 (QcstEndClusterNode) ends Cluster Resource Services on one or all nodes in the cluster.
- "List Cluster Information (QcstListClusterInfo) API" on page 37 (QcstListClusterInfo) retrieves information about a cluster.
- "List Device Domain Information (QcstListDeviceDomainInfo) API" on page 41 (QcstListDeviceDomainInfo) lists device domain information of a cluster.
- "Remove Cluster Node Entry (QcstRemoveClusterNodeEntry) API" on page 44 (QcstRemoveClusterNodeEntry) removes a node from a cluster.
- "Remove Device Domain Entry (QcstRemoveDeviceDomainEntry) API" on page 47 (QcstRemoveDeviceDomainEntry) removes a node entry from the membership list of a device domain.
- "Retrieve Cluster Information (QcstRetrieveClusterInfo) API" on page 50 (QcstRetrieveClusterInfo) retrieves information about a cluster.
- "Retrieve Cluster Resource Services Information (QcstRetrieveCRSInfo) API" on page 53 (QcstRetrieveCRSInfo) retrieves information about Cluster Resource Services parameters.
- "Start Cluster Node (QcstStartClusterNode) API" on page 57 (QcstStartClusterNode) starts Cluster Resource Services on a node in the cluster.

When a partition is detected, some APIs cannot be run in any of the partitions and some other APIs may be run in any partition. However, the action performed by the API will take effect only in the partition running the API. The restrictions for each API are:

Add Cluster Node Entry
Not allowed in any partition.

Add Device Domain Entry
Only allowed for existing device domain where all members are in the same partition.

Adjust Cluster Version
Not allowed in any partition.

Change Cluster Node Entry
To change cluster interface addresses, allowed only within the same partition. To change node status, allowed only in partition containing the non-failed nodes.

Change Cluster Resource Services
Allowed in any partition.

Create Cluster
Not allowed in any partition.

Delete Cluster
Allowed in any partition.

End Cluster Node
Allowed within the same partition as the node being ended.

List Cluster Information
Allowed in any partition.

List Device Domain Information
Allowed in any partition.

Remove Cluster Node Entry
Allowed in any partition.

Remove Device Domain Entry
Only allowed if all members are in the same partition.

Retrieve Cluster Information
Allowed in any partition.

Retrieve Cluster Resource Services Information
Allowed in any partition.

Start Cluster Node
Allowed in any partition.

Network Attributes

A network attribute is used to control cluster access within a network. The network attribute is ALWADDCLU (Allow Add to Cluster). This attribute can be set as follows:

- *NONE - The system cannot be added to any cluster.
- *ANY - The system will be added to a cluster without verification.
- *RQSAUT - The system will be added to a cluster pending successful verification through digital certificate exchange.

The default value is *NONE. If *RQSAUT is specified, » see SSL for software which must be installed on the systems. «

Cluster Node Status

Each cluster node has a status associated with it. The status of a cluster node may govern the behavior of a particular API call. See the individual API descriptions for more details. The possible values are:

1 **New.** A node has been added to the cluster membership list but the Cluster Resource Services has never been started on that node. The Cluster Resource Service data structures have not been created on the node. During a Create Cluster operation, the Cluster Resource Service data structures will be created only on the node running the Create Cluster API.

2 **Active.** The node has been started either with the Create Cluster API or Add Cluster Node Entry API with the ″Start indicator″ parameter set to 1, or with the Start Cluster Node API. Cluster Resource Services is active on the node .

3 **Remove Pending.** The node is in the process of being removed from the cluster membership list as the result of a Remove Cluster Node Entry API.

4 **Active Pending.** The node is in the process of being started either as the result of a Create Cluster API or Add Cluster Node Entry API call with the ″Start indicator″ parameter set to 1 or because of a Start Cluster Node API call. In addition, the node could have previously had a status of Partition and will change to the Active Pending status as a result of the partitions being merged.

5 **Inactive Pending.** Cluster Resource Services is in the process of ending on this node as the result of an End Cluster Node API call. The node is still in the cluster membership list.

6 **Inactive.** Cluster Resource Services has been ended on the node as the result of an End Cluster Node API call. The node is still in the cluster membership list, but is no longer communicating with other nodes in the cluster.

7 **Failed.** A previously active node has failed. A failure is defined to be a system or clustering failure detected by Cluster Resource Services.

8 **Partition.** The node is only communicating with a subset of the cluster due to a network failure

detected by Cluster Resource Services which has resulted in the loss of communications to one or more nodes in the cluster. Once the partitioned nodes have been merged back into a whole cluster, the node will change to Active status without operator intervention. Of course, any node that had a status of Failed in any partition will still have a status of Failed after the merge.

Top | "Cluster APIs," on page 1 | APIs by category

Add Cluster Node Entry (QcstAddClusterNodeEntry) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Node entry	Input	Char(*)
4	Start indicator	Input	Binary(4)
5	Format name	Input	Char(8)
6	Results information	Input	Char(30)
7	Error code	I/O	Char(*)

Service Program: QCSTCTL
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Add Cluster Node Entry (QcstAddClusterNodeEntry) API is used to add a node to the membership list of an existing cluster.

If the "Start Indicator" parameter is set to 0, the node that is being added will have a status of New and Cluster Resource Services will not be started on that node. The "Start Cluster Node (QcstStartClusterNode) API" on page 57 can be called from a program running on one of the active nodes in the cluster to start Cluster Resource Services on a node that does not have a status of Active.

If the "Start Indicator" parameter on this API is set to 1, Cluster Resource Services will be started on the node that is being added. If Cluster Resource Services is successfully started, the status for the added node will be set to Active. If the Cluster Resource Services cannot be started, the status of the added node will be set to New.

During the activation of Cluster Resource Services, the allow add to cluster (ALWADDCLU) network attribute is checked to see whether the node being added should be part of the cluster and whether to validate the cluster request through the use of X.509 digital certificates. If validation is required, the requesting node and the node being added must have the » software required for SSL installed on the systems. «

The following conditions apply to this API:

- A node cannot add itself to a cluster. It must be added from a node in the cluster that has a status of Active. If Cluster Resource Services has not been started on any of the nodes in the cluster, this API must be called from a program running on the node where the cluster was originally created, and the start indicator must be set to 0.
- The node being added to the cluster must not already be a member of this or any other cluster. A node can be a member of only one cluster.
- If the start indicator is set to 1, the node must be IP reachable (TCP/IP active and the INETD server started).
- The API will fail if any node in the cluster has a status of Partition.
- If the start indicator is set to 1, the potential node version of the node being added must be equal to the current cluster version or up to one level higher than the current cluster version. The potential node version and the current cluster version can be retrieved by using the "List Cluster Information

(QcstListClusterInfo) API" on page 37. The potential node version can also be retrieved by using the "Retrieve Cluster Information (QcstRetrieveClusterInfo) API" on page 50.

This API operates in an asynchronous mode. See "Behavior of Cluster Resource Services APIs" on page 221 for more information.

Restriction: This API cannot be called from a cluster resource group exit program.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

User Queue Authority
*OBJOPR and *ADD

User Queue Library Authority
*EXECUTE

User Queue Lock
*EXCLRD

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster to which the node is being added. It must be a valid simple name.

Node entry
INPUT; CHAR(*)

This parameter contains the information associated with the node which is being added to the cluster membership list.

Start indicator
INPUT; BINARY(4)

An indicator which specifies whether or not Cluster Resource Services is to be started on the node that is being added.

0 Cluster Resource Services will not be started on the node.
1 Cluster Resource Services will be started on the node.

Format name
INPUT; CHAR(8)

The content and format of the information supplied for the node entry. The possible format names are:

"ADDN0100 Format" on page 6 Node entry information

Results information
INPUT; CHAR(30)

A library qualified user queue name followed by a reserved field.

Library qualified user queue: A user queue, which exists on the node from which the API was called, that receives results information after the function has completed on all active nodes in the cluster. See the **Usage Notes** section of this API for a description of the data that is placed on this queue. This is a 20 character field. The first 10 characters contain the user queue name and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, and *CURLIB are not valid for the library name. The attributes of this user queue must be keyed.

Reserved. The last 10 characters of results information are reserved and must be set to hexadecimal zero.

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

ADDN0100 Format

Offset		Type	Field
Dec	**Hex**		
0	0	CHAR(8)	Node id
8	8	BINARY(4)	Offset to first cluster interface entry
12	C	BINARY(4)	Number of cluster interfaces
This field repeats for each number of cluster interfaces.		CHAR(16)	Cluster interface address

Field Descriptions

Cluster interface address. The cluster interface address is an IP address that is used by Cluster Resource Services to communicate with other nodes in the cluster. The address is in dotted decimal format and is a null-terminated string.

Note: Cluster Resource Services uses existing IP interfaces configured for an iSeries. See TCP/IP for instructions for configuring IP interfaces on the iSeries. The IP addresses defined as cluster interface addresses can be used by other applications. If an IP address is reconfigured through the TCP/IP configuration functions, the “Change Cluster Node Entry (QcstChangeClusterNodeEntry) API” on page 14 should be used to make the corresponding change to the cluster interface address. A mismatch will cause cluster errors to occur.

Node id. A valid simple name that uniquely identifies the node.

Number of cluster interfaces. The number of IP interfaces associated with a cluster node. It is limited to 1 or 2 entries only.

Offset to first cluster interface entry. The offset from the beginning of the structure to the first cluster interface address entry.

Usage Notes

Results Information User Queue. Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See “Cluster APIs Use of User Queues” on page 221 and “Using Results Information” on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue

in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF2204 D	User profile &1 not found.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPFBB01 D	Cluster already exists.
CPFBB05 D	Cluster node &1 cannot be started.
CPFBB07 D	Node &1 could not be added to cluster &2.
CPFBB11 D	Cluster node &1 already exists in cluster &2.
CPFBB12 D	Cluster node &1 in cluster &2 could not be started.
CPFBB13 D	Cluster interface address &1 already assigned to cluster node &2.
CPFBB17 D	&1 API cannot be processed in cluster &2.
CPFBB24 D	Node &1 not participating in &2 API protocol.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB46 D	Cluster Resource Services internal error.
CPFBB54 D	Node &1 not be added to the cluster &2.
CPFBB71 D	Potential node version &1 of node &2 not compatible.
CPFBB96 D	Internal device domain mismatch.
CPIBB03 I	Cluster node &1 added to cluster &2.
CPIBB05 I	Cluster node &1 started in cluster &2.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the "Usage Notes" on page 6 above.

Message ID	Message Text
CPF2113 E	Cannot allocate library &1.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C29 E	Object name &1 is not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB04 E	Number of cluster interface addresses not valid.
CPFBB07 E	Node &1 could not be added to cluster &2.
CPFBB0D E	Cluster interface address &1 specified more than once.
CPFBB11 E	Cluster node &1 already exists in cluster &2.
CPFBB13 E	Cluster interface address &1 already assigned to cluster node &2.
CPFBB17 E	&1 API cannot be processed in cluster &2.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB32 E	Attributes of user queue &1 in library &1 are not valid.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB44 E	&1 API cannot be called from a cluster resource group exit program.
CPFBB46 E	Cluster Resource Services internal error.
CPFBB55 E	Value &1 specified for start indicator not valid.

Message ID	Message Text
CPFBB57 E	Offset to cluster interface entry not valid.
TCP1901 E	Internet address &1 not valid.

API introduced: V4R4

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Add Device Domain Entry (QcstAddDeviceDomainEntry) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Device domain name	Input	Char(10)
4	Device domain entry information	Input	Char(*)
5	Format name	Input	Char(8)
6	Results information	Input	Char(30)
7	Error code	I/O	Char(*)

Service Program: QCSTDD
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Add Device Domain Entry (QcstAddDeviceDomainEntry) API is used to add a cluster node to the membership list of a device domain. There is no API to create a device domain, the device domain will be created when the first cluster node is added to it.

The following conditions apply to this API:

- This API can be called from a program running on any node in the cluster which has a status of Active.
- The node to be added and at least one current member of the device domain must be Active. On certain conditions, all current members of the device domain must be Active.
- A node can only be a member of one device domain.
- The API will fail if any member of the device domain to which the node being added has a status of Partition.
- The API will fail if it is the first node being added to the device domain and any node in the cluster has a status of Partition.

This API requires that i5/OS option 41, HA Switchable Resources, is installed and a valid license key exists on all cluster nodes that will be in the device domain.

For more information, see these Information Center topics:

- Device domains
- Resilient devices and device cluster resource groups
- Auxiliary storage pools

This API operates in an asynchronous mode. See "Behavior of Cluster Resource Services APIs" on page 221 for more information.

Restriction: This API cannot be called from a cluster resource group exit program.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

User Queue Authority
*OBJOPR and *ADD

User Queue Library Authority
*EXECUTE

User Queue Lock
*EXCLRD

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster that contains the node.

Device domain name
INPUT; CHAR(10)

The name of the device domain to which the node is being added. It must be a valid simple name. If the device domain does not currently exist, it will be created.

Device domain entry information
INPUT; CHAR(*)

Detailed information about device domain entry to be added to the device domain.

Format name
INPUT; CHAR(8)

The content and format of the device domain entry information. The possible format names are:

"ADDD0100 Format" on page 10 Node id information.

Results information
INPUT; CHAR(30)

A library qualified user queue name followed by a reserved field.

Library qualified user queue: A user queue, which exists on the node from which the API was called, that receives results information after the function has completed on all active nodes in the cluster. See the **Usage Notes** section of this API for a description of the data that is placed on this queue. This is a 20 character field. The first 10 characters contain the user queue name and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, and *CURLIB are not valid for the library name. The attributes of this user queue must be keyed.

Reserved. The last 10 characters of results information are reserved and must be set to hexadecimal zero.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Device Domain Entry Information

ADDD0100 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of data provided
4	4	CHAR(8)	Node id

Field Descriptions

Length of data provided. This is the total length of data provided (including this field) for the device domain entry information.

Node id. A unique string of characters that identifies a cluster node to be added to the device domain.

Usage Notes

Results Information User Queue Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See "Cluster APIs Use of User Queues" on page 221 and "Using Results Information" on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPFBB02 D	Cluster &1 does not exist.
CPFBB09 D	Cluster node &1 does not exist in cluster &2.
CPFBB17 D	&1 API cannot be processed in cluster &2.
CPFBB24 D	Node &1 not participating in &2 API protocol.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB46 D	Cluster Resource Services internal error.
CPFBB73 D	Cluster node &1 could not be added to device domain &2.
CPFBB74 D	Cluster node &1 already a member of device domain &2.
CPFBB78 E	Request cannot be processed in cluster &1.
CPFBB93 D	Base operating system option 41 not installed or license key not valid.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the "Usage Notes" above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C29 E	Object name &1 is not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3CF1 E	Error code parameter not valid.

Message ID	Error Message Text
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB09 E	Cluster node &1 does not exist in cluster &2.
CPFBB17 E	&1 API cannot be processed in cluster &2.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB32 E	Attributes of user queue &1 in library &1 are not valid.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB44 E	&1 API cannot be called from a cluster resource group exit program.
CPFBB46 E	Cluster Resource Services internal error.
CPFBB5F E	Field value within structure is not valid.
CPFBB70 E	API request &1 not compatible with current cluster version.
CPFBB73 E	Cluster node &1 could not be added to device domain &2.
CPFBB74 E	Cluster node &1 already a member of device domain &2.
CPFBB78 E	Request cannot be processed in cluster &1.

API introduced: V5R1

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Adjust Cluster Version (QcstAdjustClusterVersion) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Cluster version information	Input	Char(*)
4	Format name	Input	Char(8)
5	Results information	Input	Char(30)
6	Error code	I/O	Char(*)

Service Program: QCSTCTL2
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Adjust Cluster Version (QcstAdjustClusterVersion) API is used to adjust the current version of the cluster. The current cluster version is the version at which the nodes in the cluster are actively communicating with each other. In addition, this value determines what nodes can join into the cluster and the cluster's ability to use new functions supported by the nodes potential node version. It is set when the cluster is created. The version is adjusted to be one level greater than the existing value. To view the current cluster version, use the "List Cluster Information (QcstListClusterInfo) API" on page 37 or the "Retrieve Cluster Information (QcstRetrieveClusterInfo) API" on page 50.

This command will not cause the "Cluster Resource Group Exit Program" on page 185 to be called.

The following conditions apply to this API:

- This API must be called from a program running on a cluster node with a status of Active. The node must have a version of the operating system that is equal to or greater than node version of 2.

- This API cannot be used if the cluster is in a partitioned state.
- This API can only be used to adjust to a higher version. The only way to change the cluster to a lower version is to delete and recreate the cluster at the lower version.
- The cluster version cannot be set higher than the lowest potential node version in the cluster. To view the potential node versions, use the List Cluster Information or the Retrieve Cluster Information API.

This API operates in an asynchronous mode. See "Behavior of Cluster Resource Services APIs" on page 221 for more information.

Restriction: This API cannot be called from a cluster resource group exit program.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

User Queue Authority
*OBJOPR and *ADD

User Queue Library Authority
*EXECUTE

User Queue Lock
*EXCLRD

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster being adjusted.

Cluster version information
INPUT; CHAR(*)

Additional details for adjusting the cluster version. When the format name is *NONE, the cluster version information field must be CHAR(10) and filled with blanks or omitted. When the default is taken by filling in blanks or omitting this field, Cluster Resource Services adjusts the version up by 1.

Format name
INPUT; CHAR(8)

The format of the cluster version information. This field must be set to *NONE and must be left-justified.

Results information
INPUT; CHAR(30)

A library qualified user queue name followed by a reserved field.

Library qualified user queue: A user queue, which exists on the node from which the API was called, that receives results information after the function has completed on all active nodes in the cluster. See the "Usage Notes" on page 13 section of this API for a description of the data that is placed on this queue. This is a 20 character field. The first 10 characters contain the user queue name and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, and *CURLIB are not valid for the library name. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of results information are reserved and must be set to hexadecimal zero.

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See "Cluster APIs Use of User Queues" on page 221 and "Using Results Information" on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF3C4B D	Value not valid for field &1.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPFBB17 D	&1 API cannot be processed in cluster &2.
CPFBB24 D	Node &1 not participating in &2 protocol.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB2F D	Cluster version &1 cannot be adjusted.
CPFBB46 D	Cluster Resource Services internal error.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the "Usage Notes" above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3C4B E	Value not valid for field &1.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB17 E	&1 API cannot be processed in cluster &2.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB2F E	Cluster version &1 cannot be adjusted.
CPFBB32 E	Attributes of user queue &1 in library &2 are not valid.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB44 E	&1 API cannot be called from a cluster resource group exit program.
CPFBB46 E	Cluster Resource Services internal error.

API introduced: V5R1

Change Cluster Node Entry (QcstChangeClusterNodeEntry) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Node id	Input	Char(8)
4	Format name	Input	Char(8)
5	Node entry information	Input	Char(*)
6	Length of node entry information	Input	Binary(4)
7	Results information	Input	Char(30)
8	Error code	I/O	Char(*)

Service Program: QCSTCTL
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Change Cluster Node Entry (QcstChangeClusterNodeEntry) API is used to change the fields in the cluster node entry. The fields that can be changed are the cluster interface addresses defined for the node and status of the node. The node entry which is being changed may or may not have Cluster Resource Services started.

The following conditions apply to this API:

- This API must be called from a program running on a cluster node with a status of Active.
- If the cluster is in a partitioned state, this operation can only be performed within the partition running the API.
- Only one cluster interface address can be changed at a time. If the cluster is in partitioned state, the change cluster interface address is only allowed for a node within the same partition.
- To change the cluster node status, only a node that has a status of Partition or Failed can be changed and it can only be changed to Failed status.

This API operates in an asynchronous mode. See "Behavior of Cluster Resource Services APIs" on page 221 for more information.

» For "Types of Cluster Resource Groups" on page 61 cluster resource groups: «

- 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 by assigning the specified node 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 in the cluster resource group.
- If the failed node was the primary node for a cluster resource group, the first active backup will be reassigned as the new primary node. When this occurs for a device cluster resource group, ownership of the hardware will be moved to the new primary node.

» For peer model cluster resource groups, the membership status is changed to inactive. «

If an exit program is specified for the cluster resource group, it will be called with an action code of Change Node Status (20).

If a problem is detected and the API does not complete successfully, the API can be run again once the problem is corrected. Any cluster resource group that had already had the status of a node changed from Partition to Failed and the recovery domain order changed will not be affected by running this API again.

» When changing the node status to failed, the exit program for any cluster resource group which has the changing node in its recovery domain will be called with an action code of 20 (change node status). Since this is not a failover, no message is sent to the failover message queue. The cluster resource group status will remain at its current status value. «

Warning:

Using this API to change the status of a node to failed provides a way to tell Cluster Resource Services that a node has really failed. There are certain failure conditions that Cluster Resource Services cannot detect as a node failure. Rather, the problem appears to be a communication problem and the cluster looks like it has become partitioned. By telling Cluster Resource Services that a node has failed, it makes recovery from the partition state simpler » for primary-backup model cluster resource groups « since a backup node from the remaining active cluster nodes can then be assigned as the primary node. 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 allows a node in each partition to become the primary node for a cluster resource group. When two nodes think they are the primary node, data such as files or data bases could become corrupted if two different nodes are each making independent changes to copies of their files. In addition, from the cluster resource group perspective the two partitions cannot be merged back together when a node in each partition has been assigned the primary role.

Restriction: This API cannot be called from a cluster resource group exit program.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

User Queue Authority
*OBJOPR and *ADD

User Queue Library Authority
*EXECUTE

User Queue Lock
*EXCLRD

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster.

Node id
INPUT; CHAR(8)

A valid simple name that uniquely identifies the node.

Format name
INPUT; CHAR(8)

The content and format of the node entry information. The possible format names are:

"IFCA0100 Format"	Add cluster node interface
"IFCR0100 Format"	Remove cluster node interface
"IFCC0100 Format" on page 17	Replace cluster node interface
"STSC0100 Format" on page 17	Change cluster node status

Node entry information
INPUT; CHAR(*)

Detailed information about the node entry.

Length of node entry information
INPUT; BINARY(4)

The length of the node entry information.

Results information
INPUT; CHAR(30)

A library qualified user queue name followed by a reserved field.

Library qualified user queue: A user queue, which exists on the node from which the API was called, that receives results information after the function has completed on all active nodes in the cluster. See the "Usage Notes" on page 17 section of this API for a description of the data that is placed on this queue. This is a 20 character field. The first 10 characters contain the user queue name and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, and *CURLIB are not valid for the library name. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of results information are reserved and must be set to hexadecimal zero.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

IFCA0100 Format

Offset		Type	Field
Dec	Hex		
0	0	CHAR(16)	Cluster interface address

IFCR0100 Format

Offset		Type	Field
Dec	Hex		
0	0	CHAR(16)	Cluster interface address

IFCC0100 Format

Offset		Type	Field
Dec	Hex		
0	0	CHAR(16)	Old cluster interface address
16	10	CHAR(16)	New cluster interface address

STSC0100 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	New node status

Field Descriptions

Cluster interface address. The cluster interface address is an IP address that is used by Cluster Resource Services to communicate with other nodes in the cluster. The address is in dotted decimal format and is a null-terminated string.

Note: Cluster Resource Services uses existing interface addresses configured for an iSeries. See TCP/IP for instructions for configuring interface addresses on the iSeries. The IP addresses defined as cluster interface addresses can be used by other applications. If an IP address is reconfigured through the TCP/IP configuration functions, the Change Cluster Node Entry API should be used to make the corresponding change to the cluster interface address. A mismatch will cause cluster errors to occur. An interface address should not be used as a takeover IP address.

New cluster interface address. The cluster interface address which replaces the old cluster interface address. The address is in dotted decimal format and is a null-terminated string.

New node status. The new status of the node. The valid value for new node status is:

7 The node has failed.

Old cluster interface address. The cluster interface address which is being replaced. The address is in dotted decimal format and is a null-terminated string.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See "Cluster APIs Use of User Queues" on page 221 and "Using Results Information" on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Message Text
CPCBB01 C	Cluster Resource Services API &1 completed. completed.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPFBB09 D	Cluster node &1 does not exist in cluster &2.

Message ID	Message Text
CPFBB13 D	Cluster interface address &1 already assigned to cluster node &2.
CPFBB14 D	Cluster interface address &1 cannot be added for cluster node &2.
CPFBB15 D	Cluster interface address &1 cannot be removed.
CPFBB16 D	Cluster interface address &1 cannot be changed to &2.
CPFBB17 D	&1 API cannot be processed in cluster &2.
CPFBB24 D	Node &1 not participating in &2 protocol.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB46 D	Cluster Resource Services internal error.
CPFBB47 D	Cluster Resource Services ended abnormally.
CPFBB89 D	The current status of cluster node &1 cannot be changed.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the "Usage Notes" on page 17 above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB09 E	Cluster node &1 does not exist in cluster &2.
CPFBB13 E	Cluster interface address &1 already assigned to cluster node &2.
CPFBB14 E	Cluster interface address &1 cannot be added for cluster node &2.
CPFBB15 E	Cluster interface address &1 cannot be removed.
CPFBB16 E	Cluster interface address &1 cannot be changed to &2.
CPFBB17 E	&1 API cannot be processed in cluster &2.
CPFBB32 E	Attributes of user queue &1 in library &2 are not valid.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB44 E	&1 API cannot be called from a cluster resource group exit program.
CPFBB46 E	Cluster Resource Services internal error.
CPFBB56 E	Length of node entry not valid.
CPFBB5F E	Field value within structure is not valid.
CPFBB70 E	API request &1 not compatible with current cluster version.
CPFBB89 E	The current status of cluster node &1 cannot be changed.
TCP1901 E	Internet address &1 not valid.

API introduced: V4R4

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Change Cluster Resource Services (QcstChgClusterResourceServices) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Cluster resource services information	Input	Char(*)
4	Length of cluster resource services information	Input	Binary(4)
5	Format name	Input	Char(8)
6	Results information	Input	Char(30)
7	Error code	I/O	Char(*)

» Service Program: QCSTCTL2

«

Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Change Cluster Resource Services (QcstChgClusterResourceServices) API is used to tune cluster performance and configuration parameters. The API provides 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 using format CRSC0100. If an advanced level of tuning is desired, usually anticipated with the help of IBM support personnel, then individual parameters may be tuned over a predefined range of values using format CRSC0200. Example control language command source has been provided in the base operating system option 7 (Example Tools Library, QUSRTOOL). See member, TCSTINFO, in file QUSRTOOL/QATTSYSC for more information.

The default values are set on a create operation and changes must be made under the Change Cluster Resource Services API documented here. Values for current settings may be retrieved using the "Retrieve Cluster Resource Services Information (QcstRetrieveCRSInfo) API" on page 53.

The rules for merging of partitioned nodes are as follows:

- If the tuning parameters defined under the Change Cluster Resource Services API documented here match exactly in both partitions, a merge will be allowed.

This API operates in an asynchronous mode. See "Behavior of Cluster Resource Services APIs" on page 221 for more information.

Restriction: This API cannot be called from a cluster resource group exit program.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

User Queue Authority
*OBJOPR and *ADD

User Queue Library Authority
*EXECUTE

User Queue Lock
*EXCLRD

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster.

Cluster resource services information
INPUT; CHAR(*)

Detailed information about the cluster resource services.

Length of cluster resource services information
INPUT; BINARY(4)

The length of the cluster resource services information.

Format name
INPUT; CHAR(8)

The format of the Cluster Resource Services information to be changed. The possible format names are:

"CRSC0100 Format"	Automatic tuning to a level of high, low or normal heartbeat intervals and message timeout values for cluster performance and configuration parameters.
"CRSC0200 Format" on page 21	Manually tune one or more of the cluster performance and configuration parameters.

Results information
INPUT; CHAR(30)

A library qualified user queue name followed by a reserved field.

Library qualified user queue: A user queue, which exists on the node from which the API was called, that receives results information after the function has completed on all active nodes in the cluster. See the "Usage Notes" on page 25 section of this API for a description of the data that is placed on this queue. This is a 20 character field. The first 10 characters contain the user queue name and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, and *CURLIB are not valid for the library name. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of results information are reserved and must be set to hexadecimal zero.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

CRSC0100 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Configuration tuning level

CRSC0200 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(8)	Receive/send heartbeat timer ratio
8	8	BINARY(8)	Maximum retry timer ratio
16	10	BINARY(8)	Send heartbeat interval
24	18	BINARY(8)	Retry timer value
32	20	BINARY(8)	CDAT protocol timeout interval
40	28	BINARY(8)	Cluster recovery interval
48	30	BINARY(8)	Maximum retry time
56	38	BINARY(8)	Message fragment size
64	40	BINARY(8)	Send queue overflow
72	48	BINARY(8)	Number of bad messages threshold
80	50	BINARY(8)	Number of ack messages threshold
88	58	BINARY(8)	Unreachable heartbeat ack threshold
96	60	BINARY(8)	Reachable heartbeat ack threshold
104	68	BINARY(8)	Unreachable heartbeat threshold
112	70	BINARY(8)	Reachable heartbeat threshold
120	78	BINARY(8)	Delayed ack timer
128	80	BINARY(8)	Message send window
136	88	BINARY(8)	Enable multicast
144	90	BINARY(8)	Performance class
152	98	BINARY(8)	Ack remote fragments

Field Descriptions

Note: Specify -1 on any parameters that are not changed. This pertains to format CRSC0200 only.

Note: Units and ranges for the fields described here may be found in the Field Settings Range Table located at the end of this Field Descriptions section of this document.

Ack remote fragments. Provides a switch to enable or disable a cluster messaging level acknowledgment for receipt of each fragment sent to a remote cluster node. Fragments are sent by the cluster messaging service for each cluster message whose size is greater than the specified Message fragment size. Remote cluster nodes are defined to be any nodes not on the local LAN (having a network or subnet IP address other than that of the source node for the message). ACKing remote fragments may be desirable in those few cases where low bandwidth gateways, routers, or bridges exist between local and remote systems.

CDAT protocol timeout interval. The timeout value used for distributing the Cluster Destination Address Table (CDAT) and synchronizing cluster communications when doing a create cluster, add node, or start node process. As the number of nodes in the cluster increases, the time required to run this synchronizing protocol increases. This is a low level Cluster Resoure Services start-up protocol.

Cluster recovery interval. The interval at which a cluster node takes inventory of required recovery actions and attempts automatic recovery as necessary. Those items checked are:

- Unreachable alternate point-point interface addresses for remote nodes.
- Unreachable multicast IP address for the local subnet.

- Partitioned nodes.

Configuration tuning level. Provides for a simple way to set cluster performance and configuration parameters. The valid values for this field are:

1 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 are used for cluster communications performance and configuration parameters. This setting may be used to return all parameters to the original default values.

3 Adjustments are made to cluster communications to » increase « the heartbeating » frequency « and decrease the various message timeout values. With more frequent heartbeats and shorter timeout values, the cluster will be quicker to respond (more sensitive) to communications failures.

Delayed ack timer. The timer used over inbound reliable messages to force an acknowledgment for unacknowledged messages should the sender not have requested an acknowledgment over the last delayed ack time period. This timer is started on receipt of a reliable message and stopped when an acknowledgment is sent for one or more unacknowledged messages.

Enable multicast. The cluster communications infrastructure makes use of User Datagram Protocol (UDP) multicast capabilities as the preferred protocol for sending cluster management information between nodes in a cluster. Where multicast capabilities are supported by the underlying physical media, cluster communications will utilize the UDP multicast to send management messaging from a given node to all local cluster nodes supporting the same subnet address. Messages being sent to nodes on remote networks will always be sent using UDP point to point capabilities. Cluster communications does not rely on routing capability of multicast messages.

The multicast traffic supporting cluster management messaging tends by nature to be bursty. Depending on the number of nodes on a given LAN (supporting 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/second. Bursts of this nature could have a negative impact on older networking equipment. One example would be congestion problems on devices on the LAN serving as Simple Network Management Protocol (SNMP) agents which need to evaluate each and every UDP multicast packet. Some of the earlier networking equipment does not have adequate bandwidth to keep up with this type of traffic. Insure that the network administrator has reviewed the capacity of the networks to handle UDP multicast traffic to make certain that clustering will not have a negative impact on the health and performance of the networks over which it is chosen to operate.

If the network does not wish to have the more efficient multicast capabilities used, setting this field to FALSE (0) will disable the multicast capabilities of the cluster and only point to point communications will be used by the cluster messaging services.

Maximum retry time. Reliable messages are resent at exponentially increasing times should they timeout (that is, not receive a timely acknowledgment). The initial timeout value for a message is the Retry Timer Value and each successive retry builds up by a factor of 2 until the Maximum retry timer value is exceeded. For the default cases, a message would be sent, resent 1 second later, then 2 seconds, 4 seconds, and finally 8 seconds. This represents a total of 15 seconds following which attempts to use alternate internet addressing are tried with the same timer values.

Maximum retry timer ratio. Remote subnets (remote cluster nodes on another LAN/WAN/BUS supporting a different subnet interface address than the sending node) use an extended message timeout value which is based from the Maximum retry time used for local subnets (local cluster nodes supporting the same subnet interface address). For the default case, the Maximum retry time for a local multicast message would be 8 seconds and for a remote point to point message would be 8 x 8 = 64 seconds. This allows for network routing considerations.

Message fragment size. Cluster communications fragments its own messages. This fragment size should be set consistent with the physical media and routing capabilities throughout the network used for clustering. The preferred settings allow for the largest fragment size possible that does not exceed any of the hardware Maximum Transmission Units defined over the entire path so that clustering does all of the fragmentation, not the intermediary networks. The default is set to assume a minimum 1500 byte (less network header space) Ethernet environment.

Message send window. The number of messages allowed outstanding without having received an acknowledgment. The higher the number, the lower the message latency but the larger the required buffer space on a node to save inbound messages.

Number of ack messages threshold. The number of repeat messages that are received over the course of a cluster recovery interval before acknowledgments are sent to multiple source interface addresses for a given node instead of just the current primary address for each message received. While increasing the number of ACKs flowing, this reduces the message resends and latency given that an intermittent communications condition is detected. Eventually, one of the node addresses should be marked as failed and at cluster recovery time, messaging will settle back down using single acknowledgments.

Number of bad messages threshold. The number of undeliverable messages per Cluster recovery interval allowed before a failing status is assigned to a node's interface address. At this time, a secondary address (if available) is assigned to be the new primary interface address for the subject remote node.

Performance class. The requested performance characteristics of the cluster communications messaging protocol. Pacing is selectively used for sending out fragments of large messages. Messages are fragmented by the cluster messaging service at the specified message fragment size. The pacing mechanism releases a set number of fragments to the underlying physical layer, then delays, then releases a next set. This is to avoid over running slower physical media. Local here refers to nodes on a local LAN. Remote refers to messaging to cluster nodes on other than the local LAN. Valid values for the performance class are as follows:

0 Normal: Pacing applied to local and remote fragments.
1 High Throughput Local: Pacing applied to remote fragments.
2 High Throughput Local and Remote: No pacing of any fragmented messages.
3 High Throughput Remote: Pacing applied to local fragments.

Reachable heartbeat ack threshold. A node becomes reachable (formerly having been marked as unreachable) from a Cluster Communications heartbeating perspective if ″Reachable heartbeat ack threshold″ (or greater) heartbeat message ACKs are received for the last ″Reachable heartbeat threshold″ heartbeat messages sent to a node. For the default case, a node becomes reachable if 3 or more of the last four heartbeats sent to the marked unreachable node are now acknowledged.

Reachable heartbeat threshold. See Reachable heartbeat ack threshold field description.

Receive/send heartbeat timer ratio. Ratio of incoming heartbeat messages expected from a neighboring node to the number of heartbeat messages that are sent out. The send rate is always set higher to insure a neighboring node's receive heartbeat timer does not fire under normal operational circumstances.

Retry timer value. See Maximum retry time field description.

Send heartbeat interval. The interval at which a low level Cluster Communications heartbeat message is sent to a neighboring node.

Send queue overflow. The maximum number of messages that are allowed to be queued up in a Cluster Communications outbound message queue. The cluster communication send queues are distributed amongst the various groups. The larger the number, the greater the memory resources that are required

to support cluster messaging. If a send queue overflow is hit for a given group, the inability to send a message could lead to the termination of that group resulting from the lack of resources on a node.

Unreachable heartbeat ack threshold. A reachable node becomes unreachable from a cluster communications heartbeating perspective if ″Unreachable heartbeat ack threshold″ heartbeat message ACKs (or less) are received for the last ″Unreachable heartbeat threshold″ heartbeat messages sent to a node. For the default case, a node becomes unreachable if one or less of the last four heartbeats sent to the marked reachable node are acknowledged.

Unreachable heartbeat threshold. See Unreachable heartbeat ack threshold field description.

Field Settings For CRSC0200 Format

	Configuration Tuning Level			
Field	**1**	**2**	**3**	**Unit**
Receive/send heartbeat timer ratio	4	2	2	unitless
Maximum retry timer ratio	8	8	4	unitless
Send heartbeat interval	6	3	1	seconds
Retry timer value	2	1	1	seconds
CDAT protocol timeout interval	4	2	2	minutes
Cluster recovery interval	30	15	10	minutes
Maximum retry time	16	8	4	seconds
Message fragment size	1,464	1,464	1,464	bytes
Send queue overflow	1,024	1,024	1,024	messages
Number of bad messages threshold	5	3	2	messages
Number of ack messages threshold	20	10	5	messages
Unreachable heartbeat ack threshold	1	1	1	messages
Reachable heartbeat ack threshold	3	3	3	messages
Unreachable heartbeat threshold	4	4	4	messages
Reachable heartbeat threshold	4	4	4	messages
Delayed ack timer	300	100	50	milliseconds
Message send window	2	2	2	messages
Enable multicast	TRUE(1)	TRUE(1)	TRUE(1)	unitless
Performance class	2	2	2	unitless
Ack remote fragments	FALSE(0)	FALSE(0)	FALSE(0)	unitless

Field Settings Range

Field	Minimum	Default	Maximum	Unit
Receive/send heartbeat timer ratio	2	2	4	unitless
Maximum retry timer ratio	1	8	8	unitless
Send heartbeat interval	1	3	10	seconds
Retry timer value	1	1	4	seconds
CDAT protocol timeout interval	1	2	5	minutes
Cluster recovery interval	5	15	60	minutes

Field	Minimum	Default	Maximum	Unit
Maximum retry time	4	8	16	seconds
Message fragment size	540	1,464	32,500	bytes
Send queue overflow	512	1,024	4,096	messages
Number of bad messages threshold	2	3	50	messages
Number of ack messages threshold	2	10	25	messages
Unreachable heartbeat ack threshold	1	1	m	messages
Reachable heartbeat ack threshold	1	3	n	messages
Unreachable heartbeat threshold	2	4	16	messages
Reachable heartbeat threshold	2	4	16	messages
Delayed ack timer	50	100	300	milliseconds
Message send window	1	2	8	messages
Enable multicast	FALSE(0)	TRUE(1)	TRUE(1)	unitless
Performance class	0	2	3	unitless
Ack remote fragments	FALSE(0)	FALSE(0)	TRUE(1)	unitless

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See "Cluster APIs Use of User Queues" on page 221 and "Using Results Information" on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPFBB24 D	Node &1 not participating in &2 API protocol.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB46 D	Cluster Resource Services internal error.
CPFBB4D D	Cluster Resource Services cannot process the request.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the "Usage Notes" above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.

Message ID	Error Message Text
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB32 E	Attributes of user queue &1 in library &2 are not valid.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB44 E	&1 API cannot be called from a cluster resource group exit program.
CPFBB46 E	Cluster Resource Services internal error.
CPFBB5F E	Field value within structure is not valid.
CPFBB70 E	API request &1 not compatible with current cluster version.
CPFBB86 E	Length specified in parameter &1 not valid.

API introduced: V5R1

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Create Cluster (QcstCreateCluster) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Cluster membership information	Input	Char(*)
4	Number of cluster membership entries	Input	Binary(4)
5	Start indicator	Input	Binary(4)
6	Format name	Input	Char(8)
7	Results information	Input	Char(30)
8	Error code	I/O	Char(*)

Service Program: QCSTCTL
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Create Cluster (QcstCreateCluster) API is used to create a new cluster of one or more nodes. Each node specified on the "Cluster membership information" parameter will be placed in the cluster membership list.

If the "Start indicator" parameter is set to 0, each node that is being added will have a status of New and Cluster Resource Services will not be started on any node. In order to start Cluster Resource Services, the "Start Cluster Node (QcstStartClusterNode) API" on page 57 must be called from a program running on the node that ran the Create Cluster API. The Start Cluster Node API may be used to start nodes in the cluster membership list.

If the "Start indicator" parameter is set to 1, the cluster can contain only one node. Cluster Resource Services will be started on the node being defined. If Cluster Resource Services is successfully started, the status for the node will be set to Active. If Cluster Resource Services is not successfully started, the status of the node remains New. If a list of nodes is specified, the start indicator is ignored.

If the NODE0100 format is chosen, the current cluster version will be set equal to the requesting node's potential node version.

After Cluster Resource Services has been started on the original node, additional nodes can only be started by calling the Start Cluster Node API on the original node. If Cluster Resource Services is active on more than one node, additional nodes may be started by calling the Start Cluster Node API on any node that has a status of Active.

Once the cluster has been created, the "Add Cluster Node Entry (QcstAddClusterNodeEntry) API" on page 4 can be used to add additional nodes to the cluster membership list. The Add Cluster Node Entry API can be called from a program running on any node in the cluster that has a status of Active or from the node on which the cluster was originally created.

The following conditions apply to this API:

- A node can be a member of only one cluster.
- At least one node must be specified in the cluster membership list. A cluster cannot be created with an empty membership list.

This API operates in an asynchronous mode. See "Behavior of Cluster Resource Services APIs" on page 221 for more information.

Restriction: This API cannot be called from a cluster resource group exit program.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

User Queue Authority
*OBJOPR and *ADD

User Queue Library Authority
*EXECUTE

User Queue Lock
*EXCLRD

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster which is being created. This must be a valid simple name.

Cluster membership information
INPUT; CHAR(*)

This parameter contains information about the cluster and the list of nodes which will be placed in the cluster membership list.

Number of cluster membership entries
INPUT; BINARY(4)

The number of nodes in the cluster membership array. Must be greater than or equal to 1 and less than or equal to 128.

Start indicator
INPUT; BINARY(4)

An indicator which specifies whether or not Cluster Resource Services is to be started on the node being defined.

0 Cluster Resource Services will not be started on any node.
1 Cluster Resource Services will be started on the node.

Format name
INPUT; CHAR(8)

The content and format of the information supplied in the cluster membership information parameter. The possible format names are:

"NODE0100 Format"	Cluster membership information
"NODE0200 Format" on page 29	Cluster membership information plus additional information

Results information
INPUT; CHAR(30)

A library qualified user queue name followed by a reserved field.

Library qualified user queue: A user queue, which exists on the node from which the API was called, that receives results information after the function has completed on all active nodes in the cluster. See the "Usage Notes" on page 30 section of this API for a description of the data that is placed on this queue. This is a 20 character field. The first 10 characters contain the user queue name and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, and *CURLIB are not valid for the library name. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of results information are reserved and must be set to hexadecimal zero.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code parameter.

NODE0100 Format

Note: These fields are repeated for each node entry.

Offset			
Dec	**Hex**	**Type**	**Field**
		BINARY(4)	Length of node entry
		CHAR(8)	Node id
		BINARY(4)	Offset to first cluster interface entry
		BINARY(4)	Number of cluster interfaces
This field is repeated for each cluster interface.		CHAR(16)	Cluster interface address

NODE0200 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Target cluster version
4	4	BINARY(4)	Offset to first node entry
8	8	BINARY(4)	Offset to additional fields
12	C	BINARY(4)	Length of additional fields
Note: These fields are repeated for each node entry.			
		BINARY(4)	Length of node entry
		CHAR(8)	Node id
		BINARY(4)	Offset to first cluster interface entry
		BINARY(4)	Number of cluster interfaces
This field is repeated for each cluster interface.		CHAR(16)	Cluster interface address

Field Descriptions

Cluster interface address. The cluster interface address is an IP address that is used by Cluster Resource Services to communicate with other nodes in the cluster. The address is in dotted decimal format and is a null-terminated string.

Note: Cluster Resource Services uses existing IP interfaces configured for an iSeries. See TCP/IP for instructions for configuring IP interfaces on the iSeries. The IP addresses defined as cluster interface addresses can be used by other applications. If an IP address is reconfigured through the TCP/IP configuration functions, the "Change Cluster Node Entry (QcstChangeClusterNodeEntry) API" on page 14 should be used to make the corresponding change to the cluster interface address. A mismatch will cause cluster errors to occur.

Length of additional fields. The length of the additional fields. This must be set to hexadecimal zeros. It will be used in a future release.

Length of node entry. The length of the node entry.

Node id. A simple valid name that uniquely identifies a node.

Number of cluster interfaces. The number of IP interfaces that are to be used by Cluster Resource Services. It is limited to 1 or 2 entries only.

Offset to additional fields. The offset from the beginning of the structure to the additional fields. This must be set to hexadecimal zeros. It will be used in a future release.

Offset to first cluster interface entry. The offset from the beginning of the structure to the first cluster interface entry.

Offset to first node entry. The offset from the beginning of the structure to the first node entry.

Target cluster version. The version the cluster will use in conversation with the other nodes in the cluster. This also determines the potential node version of the nodes allowed to join the cluster. The following possible values are based on the node originating the request.

0 The cluster will communicate at the requesting node's potential node version. In addition, nodes with a potential node version less than the requesting node will not be allowed to join the cluster.

-1 The cluster will communicate at the requesting node's potential node version minus 1. This allows nodes at a previous potential node version to join the cluster. However, no new cluster function on the node which has a newer version of the system software will be allowed to be used.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See "Cluster APIs Use of User Queues" on page 221 and "Using Results Information" on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPFBB05 D	Cluster node &1 cannot be started.
CPFBB10 D	Specified cluster interface not defined on this system.
CPFBB12 D	Cluster node &1 in cluster &2 could not be started.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB46 D	Cluster Resource Services internal error.
CPIBB01 I	Cluster &1 created.
CPIBB03 I	Cluster node &1 added to cluster &2.
CPIBB05 I	Cluster node &1 started in cluster &2.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the "Usage Notes" above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C29 E	Object name &1 is not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB01 E	Cluster already exists.
CPFBB03 E	Number of cluster node entries not valid.
CPFBB04 E	Number of cluster interface addresses not valid.
CPFBB0C E	Cluster node ID &1 specified more than once.
CPFBB0D E	Cluster interface address &1 specified more than once.
CPFBB32 E	Attributes of user queue &1 in library &2 are not valid.
CPFBB39 E	Current user does not have IOSYSCFG special authority.

Message ID	Error Message Text
CPFBB44 E	&1 API cannot be called from a cluster resource group exit program.
CPFBB46 E	Cluster Resource Services internal error.
CPFBB55 E	Value &1 specified for start indicator not valid.
CPFBB56 E	Length of node entry not valid.
CPFBB57 E	Offset to cluster interface entry not valid.
CPFBB5F E	Field value within structure is not valid.
TCP1901 E	Internet address &1 not valid.

API introduced: V4R4

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Delete Cluster (QcstDeleteCluster) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Results information	Input	Char(30)
4	Error code	I/O	Char(*)

Service Program: QCSTCTL
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Delete Cluster (QcstDeleteCluster) API is used to delete a cluster created by the "Create Cluster (QcstCreateCluster) API" on page 26 API. All cluster resource group (CRG) objects and device domains associated with the cluster are deleted, Cluster Resource Services is ended on each node in the cluster membership list, and the cluster is deleted.

The following conditions apply to this API:

- The Delete Cluster API must be called from a program running on a node in the cluster.
- If the API is initiated from a cluster node with a status of Active, all active cluster nodes will be removed from the cluster, and the cluster resource group objects associated with the cluster will be deleted. Cluster resource group objects on nodes with a status of Inactive or Failed will not be deleted.
- If the API is initiated from a cluster node with a status of Failed or Inactive, only that node is removed from the cluster and cluster resource group objects on that node are deleted.
- Cluster Resource Group Manager will call cluster resource group exit programs with an action code of Delete (7) or Delete Command(14) if Cluster Resource Services is not active on the node where the API is run.
- This API may be called when the cluster is in a partitioned state. In this case, the delete operation will only be performed within the partition running the API.
- A node which was a member of a device domain has internal information related to auxiliary storage pools such as disk unit numbers or virtual memory addresses. After a cluster is deleted, this internal information persists until the node is IPLed. If the cluster is deleted, the node must be IPLed before the node can become a member of another device domain.

This API operates in an asynchronous mode. See "Behavior of Cluster Resource Services APIs" on page 221 for more information.

Restriction: This API cannot be called from a cluster resource group exit program.

Authorities and Locks

This API is shipped with *EXCLUDE public authority.

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

User Queue Authority
*OBJOPR and *ADD

User Queue Library Authority
*EXECUTE

User Queue Lock
*EXCLRD

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster which is being deleted. This must be a valid simple name.

Results information
INPUT; CHAR(30)

A library qualified user queue name followed by a reserved field.

Library qualified user queue: A user queue, which exists on the node from which the API was called, that receives results information after the function has completed on all active nodes in the cluster. See the "Usage Notes" section of this API for a description of the data that is placed on this queue. This is a 20 character field. The first 10 characters contain the user queue name and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, and *CURLIB are not valid for the library name. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of results information are reserved and must be set to hexadecimal zero.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See "Cluster APIs Use of User Queues" on page 221 and "Using Results Information" on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.

Message ID	Message Text
CPF18BA D	Error occurred with subsystem.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPFBB24 D	Node &1 is not participating in &2 API protocol
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB46 D	Cluster Resource Services internal error.
CPIBB02 I	Cluster &1 deleted from node &2.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the "Usage Notes" on page 32 above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF3C1E E	Required parameter &1 omitted.
CPF3C39 E	Value for reserved field not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB32 E	Attributes of user queue &1 in library &2 are not valid.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB44 E	&1 API cannot be called from a cluster resource group exit program.
CPFBB46 E	Cluster Resource Services internal error.

API introduced: V4R4

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

---

End Cluster Node (QcstEndClusterNode) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Node entry	Input	Char(*)
4	Option	Input	Binary(4)
5	Format name	Input	Char(8)
6	Results information	Input	Char(30)
7	Error code	I/O	Char(*)

Service Program: QCSTCTL
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The End Cluster Node (QcstEndClusterNode) API is used to end Cluster Resource Services on one or all the nodes in the membership list of an existing cluster. The status of each node that is ended is set to

Inactive. In order to restart Cluster Resource Services on nodes that have been ended, the "Start Cluster Node (QcstStartClusterNode) API" on page 57 is used.

When a node in the cluster is ended, it is not removed from the cluster membership list.

If all the nodes in the cluster are being ended, cluster resource group exit programs will not be called with an indication to failover.

The following conditions apply to this API:

- The node being ended must be ACTIVE.
- This API can be called from a program running on the node which is to be ended, or it can be called from a program running on any node in the cluster which has a status of Active.
- The cluster resource group exit program on the node being ended will be called with an action code of End Node (16). The exit program on all other nodes in the recovery domain will be called with an action code of Failover (9).
- If this API is called when the cluster is partitioned, only nodes in the partition running the API will process the request.
- The recovery domain of cluster resource groups on the node that had ended will indicate a node status of Active even though the node is Inactive. For all the other nodes in the recovery domain, the status of the node will be Inactive. If the node being ended is the primary node for an active device cluster resource group, ownership of the hardware associated with the cluster resource group will be moved to a backup node. If the cluster resource group is not active, there are no backup nodes, or all backup nodes are either inactive or in a different cluster partition, the ownership of the hardware is left with the node being ended.

This API operates in an asynchronous mode. See "Behavior of Cluster Resource Services APIs" on page 221 for more information.

Restriction: This API cannot be called from a cluster resource group exit program.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

User Queue Authority
*OBJOPR and *ADD

User Queue Library Authority
*EXECUTE

User Queue Lock
*EXCLRD

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster that contains the node or nodes being ended.

Node entry
INPUT; CHAR(*)

This parameter contains the node id being ended. It is ignored if the special value *ALL is specified as the node id.

Option
INPUT; BINARY(4)

The method used to end the node:

0 Immediate. The request to end replication for all CRGs on the node will be processed immediately.
1 Controlled. Pending CRG actions will complete before the request to end replication is processed.

Format name
INPUT; CHAR(8)

The content and format of the information supplied in the node entry array. The possible format names are:

"ENDN0100 Format" Node id information

Results information
INPUT; CHAR(30)

A library qualified user queue name followed by a reserved field.

Library qualified user queue: A user queue, which exists on the node from which the API was called, that receives results information after the function has completed on all active nodes in the cluster. See the "Usage Notes" on page 36 section of this API for a description of the data that is placed on this queue. This is a 20 character field. The first 10 characters contain the user queue name and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, and *CURLIB are not valid for the library name. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of results information are reserved and must be set to hexadecimal zero.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

ENDN0100 Format

Offset		Type	Field
Dec	Hex		
0	0	CHAR(8)	Node id

Field Descriptions

Node id.

A valid simple name that uniquely identifies a node. A special value of *ALL can be specified to end all nodes in a cluster. The special value must be left-justified.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See "Cluster APIs Use of User Queues" on page 221 and "Using Results Information" on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF18BA D	Error occurred with subsystem.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPFBB09 D	Cluster node &1 does not exist in cluster &2.
CPFBB17 D	&1 API cannot be processed in cluster &2.
CPFBB1C D	Cluster node &1 in cluster &2 cannot be ended.
CPFBB24 D	Node &1 not participating in &2 protocol.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB46 D	Cluster Resource Services internal error.
CPIBB06 I	Cluster node &1 ended in cluster &2.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the "Usage Notes" above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9804 E	Object &2 in library &3 damaged.
CF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB09 E	Cluster node &1 does not exist in cluster &2.
CPFBB17 E	&1 API cannot be processed in cluster &2.
CPFBB1C E	Cluster node &1 in cluster &2 already ended.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB32 E	Attributes of user queue &1 in library &1 are not valid.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB44 E	&1 API cannot be used within a cluster resource group exit program.
CPFBB46 E	Cluster Resource Services internal error.
CPFBB59 E	Value &1 specified for option not valid.

API introduced: V4R4

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

List Cluster Information (QcstListClusterInfo) API

Required Parameter Group:

1	Qualified user space name	Input	Char(20)
2	Cluster name	Input	Char(10)
3	Format name	Input	Char(8)
4	Node id	Input	Char(8)
5	Error code	I/O	Char(*)

Service Program: QCSTCTL1
Default Public Authority: *USE
Threadsafe: Yes

The List Cluster Information (QcstListClusterInfo) API is used to retrieve information about a cluster. It must be called from a program running on one of the nodes in the cluster. The information returned may not be current if this API is called from a program running on a node that has a status of Inactive or Failed. This API may be called from a cluster resource group exit program.

Authorities and Locks

User Space Authority
*CHANGE

User Space Library Authority
*EXECUTE

User Queue Lock
*EXCLRD

Required Parameter Group

Qualified user space name
OUTPUT; CHAR(20)

The user space that receives the information and the library in which it is located. The first 10 characters contain the user space name, and the second 10 characters contain the library name. No special values (QTEMP, *CURLIB, or *LIBL) can used for the library name.

Cluster name
INPUT; CHAR(10)

The name of the cluster for which information is retrieved.

Format name
INPUT; CHAR(8)

The format of the information to be returned. Supported format names are:

"LCTI0100 Format" on page 39	Returns information about a specific node or all nodes in the cluster and additional information about the cluster.

Node id
INPUT; CHAR(8)

A valid simple name that uniquely identifies a node. *ALL special value can be used to return information about all nodes in the cluster. The *ALL special value must be left-justified.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Format of Generated Lists

The cluster information list consists of:

- A user space
- A generic header
- A header section
- An input parameter section
- A list data section
  - LCTI0100

For details about the user area and generic header, see User Space Format for List APIs. For detailed descriptions of the fields in the list returned, see "Field Descriptions" on page 39.

When you retrieve list entry information from a user space, you must not use the entry size returned in the generic header. Each entry may have a different size. The size of each entry may be padded at the end.

Input Parameter Section

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	User space name
10	A	CHAR(10)	User space library name
20	14	CHAR(10)	Cluster name
30	1E	CHAR(8)	Format name
38	26	CHAR(8)	Node id

Header Section

Global information about the cluster.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(1)	Information status
1	1	CHAR(3)	Reserved
4	4	BINARY(4)	Current cluster version
8	8	BINARY(4)	Current cluster version modification level

LCTI0100 Format

Offset		Type	Field
Dec	Hex		
These fields are repeated for each node entry returned.		Array(*) of CHAR(*)	Node entry array
		BINARY(4)	Length of node entry
		CHAR(8)	Node id
		BINARY(4)	Node status
		BINARY(4)	Offset to first cluster interface entry
		BINARY(4)	Number of cluster interfaces
		BINARY(4)	Potential node version
		BINARY(4)	Potential node version modification level
		CHAR(10)	Device domain name
		Array(*) of CHAR(16)	Cluster interface entry array
These fields are repeated for each cluster interface entry returned.		CHAR(16)	Cluster interface address

Field Descriptions

Cluster interface address. The cluster interface address is an IP address which is used by Cluster Resource Services to communicate with other nodes in the cluster. It is returned as a null-terminated string and represented in dotted decimal format.

Cluster interface address array. Array of cluster interface addresses in use by each node in the node entry array.

Cluster name. The name of the cluster for which information is retrieved.

Current cluster version. The version at which the nodes in the cluster are actively communicating with each other. This value in conjunction with the potential node version determines what nodes can join in the cluster. This value also determines the cluster's ability to use new functions supported by the node's potential node version. It is set when the cluster is created and can be changed by the "Adjust Cluster Version (QcstAdjustClusterVersion) API" on page 11.

Current cluster version modification level. The modification level of the current cluster version. The modification level further identifies the version at which the nodes in the cluster communicate. It is updated when code changes that impact the version are applied to the system.

Device domain name. The name of the device domain that this node belongs to. This field will contain hexadecimal zeros if the node does not belong to a device domain.

Format name. The content and format of the information returned in the user space.

Information status. Indicates the consistency of the retrieved information.

0 The information is consistent for all active nodes in the cluster.

1 The information retrieved from the node running the API may not be consistent with all active nodes in the cluster. In order to obtain consistent information:

- Call this API on an active node in the cluster.
- Start Cluster Resource Services on the node and call the API again.

Length of node entry. The length of the node entry.

Node entry array. Array of cluster nodes for which information is being returned.

Node id. A valid simple name that uniquely identifies a node.

Node status. The status of the node in the cluster. See "Cluster Node Status" on page 3 for the possible values and definitions of the status.

Number of cluster interfaces. The number of IP interfaces used by the node for Cluster Resource Services.

Offset to first cluster interface entry. The offset from the beginning of the user space to the first cluster interface entry.

Potential node version. The version at which the node is capable of communicating with the other nodes in the cluster. This is the value associated with the cluster code installed on the node. It will be used to determine if the node can join a cluster. If communications have not yet been established with the node (status of New), then the potential node version will be reported as 0.

Potential node version modification level. The modification level of the potential node version. The modification level further identifies the version at which the node is capable of communicating with the other nodes in the cluster. It is updated when code changes that impact the version are applied to the system.

Reserved. The field will contain hexadecimal zeroes.

User space name. The name of the user space.

User space library name. The name of the library in which the user space resides. No special values are supported for library name.

Error Messages

Messages that are delivered through the error code parameter are listed here.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.

Message ID	Error Message Text
CPFBB09 E	Cluster node &1 does not exist in cluster &2.
CPFBB38 E	Library name &1 not allowed for this request.

API introduced: V4R4

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

---

List Device Domain Information (QcstListDeviceDomainInfo) API

Required Parameter Group:

1	Qualified user space name	Input	Char(20)
2	Cluster name	Input	Char(10)
3	Device domain name	Input	Char(10)
4	Format name	Input	Char(8)
5	Error code	I/O	Char(*)

Service Program: QCSTCTL1
Default Public Authority: *USE
Threadsafe: Yes

The List Device Domain Information (QcstListDeviceDomainInfo) API is used to retrieve information about a device domain or all device domain names in a cluster. It must be called from a program running on one of the nodes in the cluster. The information returned may not be current if this API is called from a program running on a node where cluster resource services is not active. This API may be called from a cluster resource group exit program.

Authorities and Locks

User Space Authority
*CHANGE

User Space Library Authority
*EXECUTE

User Queue Lock
*EXCLRD

Required Parameter Group

Qualified user space name
OUTPUT; CHAR(20)

The user space that receives the information and the library in which it is located. The first 10 characters contain the user space name, and the second 10 characters contain the library name. Special values are not allowed to be specified for the library name. QTEMP, *LIBL and *CURLIB are not valid for the library name.

Cluster name
INPUT; CHAR(10)

The name of the cluster.

Device domain name
INPUT; CHAR(10)

The name of the device domain for which information is retrieved. *ALL special value must be used if LDDI0200 is specified as the format name. The *ALL special value must be left-justified.

Format name

INPUT; CHAR(8)

The format of the information to be returned. Supported format names are:

"LDDI0100 Format" on page 43	Returns information about all nodes in the device domain.
"LDDI0200 Format" on page 43	Returns information about all device domain names in the cluster.

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Format of Generated Lists

The cluster information list consists of:

- A user space
- A generic header
- An input parameter section
- A header section
- A list data section
  - LDDI0100 format
  - LDDI0200 format

For details about the user area and generic header, see User Space Format for List APIs. For detailed descriptions of the fields in the list returned, see "Field Descriptions" on page 43.

Input Parameter Section

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	CHAR(10)	User space name
10	A	CHAR(10)	User space library name
20	14	CHAR(10)	Cluster name
30	1E	CHAR(10)	Device domain name
40	28	CHAR(8)	Format name

Header Section

Global information about the cluster.

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	CHAR(1)	Information status

LDDI0100 Format

Offset		Type	Field
Dec	Hex		
These fields are repeated for each node entry returned.		CHAR(8)	Node id
		BINARY(4)	Node status

LDDI0200 Format

Offset		Type	Field
Dec	Hex		
This field is repeated for each device domain entry returned.		CHAR(10)	Device domain name

Field Descriptions

Cluster name. The name of the cluster.

Device domain name. The name of the device domain.

Format name. The content and format of the information returned in the user space.

Information status. Indicates the consistency of the retrieved information.

0 The information is consistent for all active nodes in the cluster.
1 The information retrieved from the node running the API may not be consistent with all active nodes in the cluster. In order to obtain consistent information:

- Call this API on an active node in the cluster.
- Start Cluster Resource Services on the node and call the API again.

Node id. A valid simple name that uniquely identifies a node.

Node status. The status of the node in the cluster. See "Cluster Node Status" on page 3 for the possible values and definitions of the node status.

User space library name. The name of the library in which the user space resides. No special values are supported for library name. *LIBL and *CURLIB are not valid for the library name.

User space name. The name of the user space.

Error Messages

Messages that are delivered through the error code parameter are listed here.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C4B E	Value not valid for field &1.

Message ID	Error Message Text
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB38 E	Library name &1 not allowed for this request.
CPFBB70 E	API request &1 not compatible with current cluster version.
CPFBB77 E	Device domain &1 does not exist in cluster &2.

API introduced: V5R1

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Remove Cluster Node Entry (QcstRemoveClusterNodeEntry) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Node entry	Input	Char(*)
4	Format name	Input	Char(8)
5	Results information	Input	Char(30)
6	Error code	I/O	Char(*)

Service Program: QCSTCTL
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Remove Cluster Node Entry (QcstRemoveClusterNodeEntry) API is used to remove a node from a cluster. The node specified will be removed from the cluster membership list and will no longer be considered a member of the cluster. The node will also be removed from the device domain it belongs to. The cluster resource group (CRG) objects on the node being removed are deleted only if the node has a status of Active or if the program that calls this API is running on the node that is being removed.

The following conditions apply to this API:

- A node can be removed regardless of its status.
- If this API is called from a program running on a node with a status of Active, any node in the cluster can be removed.
- If this API is called from a program running on a node where cluster resource services is Inactive, only the node running the API can be removed.
- If all of the nodes in the cluster have a status of New, this API can only be called from a program running on the node where the cluster was originally created.
- To remove a node that is not Active, this API should be called from a program running on the node being removed and on a node in the cluster that is Active.
- There must be more than one node in the membership list.
- If the node being removed is Active, the cluster resource group exit program will be passed an action code of Remove Node (12). The exit program on all other nodes in the recovery domain of the cluster resource group will be passed an action code of Failover (9). If the node being removed is the primary

node for a device cluster resource group, ownership of the hardware associated with the cluster resource group will be moved to a backup node. If there are no backup nodes or all the backup nodes are either inactive or in a different cluster partition, ownership of the hardware is left with the node being removed.

- If the node being removed is Inactive, the cluster resource group exit program will be passed an action code of Remove Node (12) on all nodes in the recovery domain. Ownership of hardware associated with a device cluster resource group will not be changed but will remain with the node being removed.
- If the node being removed is Inactive, the cluster resource group exit program will be called with an action code of Delete Command (14) on the node being removed if the API is run on the node being removed.
- If the node being removed is a member of a device domain and it later will be added back to a cluster, the node most likely has to be IPLed before it can be added to any device domain. One example of this situation would be if a device description for an auxiliary storage pool has been varied on since the last IPL.

This API operates in an asynchronous mode. See "Behavior of Cluster Resource Services APIs" on page 221 for more information.

Restriction: This API cannot be called from a cluster resource group exit program.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

User Queue Authority
*OBJOPR and *ADD

User Queue Library Authority
*EXECUTE

User Queue Lock
*EXCLRD

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster from which the node will be removed.

Node entry
INPUT; CHAR(*)

This parameter contains the information associated with the node which is being removed from the cluster membership list. The size of this parameter is implied by the format name.

Format name
INPUT; CHAR(8)

The content and format of the information supplied in the node entry. The possible format names are:

"RMVN0100 Format" on page 46 Node id information

Results information

INPUT; CHAR(30)

A library qualified user queue name followed by a reserved field.

Library qualified user queue: A user queue, which exists on the node from which the API was called, that receives results information after the function has completed on all active nodes in the cluster. See the "Usage Notes" section of this API for a description of the data that is placed on this queue. This is a 20 character field. The first 10 characters contain the user queue name and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, and *CURLIB are not valid for the library name. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of results information are reserved and must be set to hexadecimal zero.

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

RMVN0100 Format

Offset		Type	Field
Dec	Hex		
0	0	CHAR(8)	Node id

Field Descriptions

Node id. A unique string of characters that identifies a node.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See "Cluster APIs Use of User Queues" on page 221 and "Using Results Information" on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF18BA D	Errors occurred with subsystem.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPFBB09 D	Cluster node &1 does not exist in cluster &2.
CPFBB24 D	Node &1 not participating in &2 protocol.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB46 D	Cluster Resource Services internal error.
CPFBB58 D	Cluster node &1 cannot be removed from cluster &2.
CPIBB04 I	Cluster node &1 removed from cluster &2.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the "Usage Notes" on page 46 above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB09 E	Cluster node &1 does not exist in cluster &2.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB32 E	Attributes of user queue &1 in library &1 are not valid.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB44 E	&1 API cannot be used within a cluster resource group exit program.
CPFBB46 E	Cluster Resource Services internal error.
CPFBB58 E	Cluster node &1 cannot be removed from cluster &2.

API introduced: V4R4

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

---

Remove Device Domain Entry (QcstRemoveDeviceDomainEntry) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Device domain name	Input	Char(10)
4	Device domain entry information	Input	Char(*)
5	Format name	Input	Char(8)
6	Results information	Input	Char(30)
7	Error code	I/O	Char(*)

Service Program: QCSTDD
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Remove Device Domain Entry (QcstRemoveDeviceDomainEntry) API is used to remove a cluster node from a device domain.

The following conditions apply to this API:

- The API will fail if the node to be removed is in the recovery domain of any device cluster resource group.
- The node to be removed and at least one other member of the device domain must be ACTIVE. On certain conditions, all current members of the device domain must be active.

- A node which has been removed from a device domain will most likely need to be IPLed before it can be added to any device domain. One example of this situation would be if a device description for an auxiliary storage pool has been varied on since the last IPL.
- This API can be called from a program running on any node in the cluster which has a status of Active.
- The API will fail if any member of the device domain from which the node being removed has a status of Partition.

For more information, see these Information Center topics:

- Device domains
- Resilient devices and device cluster resource groups
- Auxiliary storage pools

This API operates in an asynchronous mode. See "Behavior of Cluster Resource Services APIs" on page 221 for more information.

Restriction: This API cannot be called from a cluster resource group exit program.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

User Queue Authority
*OBJOPR and *ADD

User Queue Library Authority
*EXECUTE

User Queue Lock
*EXCLRD

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster that contains the node.

Device domain name
INPUT; CHAR(10)

The name of the device domain from which the node is to be removed.

Device domain entry information
INPUT; CHAR(*)

Detailed information about device domain entry to be removed from the device domain.

Format name
INPUT; CHAR(8)

The content and format of the device domain entry information. The possible format names are:

"RMVD0100 Format" on page 49 Node id information.

Results information
INPUT; CHAR(30)

A library qualified user queue name followed by a reserved field.

Library qualified user queue: A user queue, which exists on the node from which the API was called, that receives results information after the function has completed on all active nodes in the cluster. See the "Usage Notes" section of this API for a description of the data that is placed on this queue. This is a 20 character field. The first 10 characters contain the user queue name and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, and *CURLIB are not valid for the library name. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of results information are reserved and must be set to hexadecimal zero.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Device Domain Entry Information

RMVD0100 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of data provided
4	4	CHAR(8)	Node id

Field Descriptions

Length of data provided. This is the total length of data provided (including this field) for the device domain entry information. The value must equal the length of the RMVD0100 format.

Node id. A unique string of characters that identifies a cluster node to be removed from the device domain.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See "Cluster APIs Use of User Queues" on page 221 and "Using Results Information" on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPCBB02 D	Cluster &1 does not exist.
CPFBB09 D	Cluster node &1 does not exist in cluster &2.
CPFBB24 D	Node &1 not participating in &2 API protocol.
CPFBB2D D	Timeout detected while waiting for a response.

Message ID	Message Text
CPFBB46 D	Cluster Resource Services internal error.
CPFBB75 D	Cluster node &1 not a member of device domain &2.
CPFBB76 D	Cluster node &1 cannot be removed from device domain &2.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the **Usage Notes** above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C29 E	Object name &1 is not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB09 E	Cluster node &1 does not exist in cluster &2.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB32 E	Attributes of user queue &1 in library &1 are not valid.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB44 E	&1 API cannot be called from a cluster resource group exit program.
CPFBB46 E	Cluster Resource Services internal error.
CPFBB5F E	Field value within structure is not valid.
CPFBB70 E	API request &1 not compatible with current cluster version.
CPFBB75 E	Cluster node &1 not a member of device domain &2.
CPFBB76 E	Cluster node &1 cannot be removed from device domain &2.

API introduced: V5R1

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Retrieve Cluster Information (QcstRetrieveClusterInfo) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Error code	I/O	Char(*)

Service Program: QCSTCTL1
Default Public Authority: *USE
Threadsafe: Yes

The Retrieve Cluster Information (QcstRetrieveClusterInfo) API retrieves information about the clustering environment on a requesting node. The requesting node does not need to be active in the cluster to

retrieve the information. However, some of the information will not be returned if the requesting node is not currently a member of a cluster and the requesting node was never activated. This API may be called from a cluster resource group exit program.

Authorities and Locks

None

Required Parameter Group

Receiver variable
OUTPUT; CHAR(*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

Length of receiver variable
INPUT; BINARY(4)

The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

Format name
INPUT; CHAR(8)

The content and format of the information that is returned. The possible format names are as follows:

"RCLI0100 Format" Cluster information

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

RCLI0100 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	Cluster name
18	12	CHAR(8)	Requesting node id
26	1A	CHAR(2)	Reserved
28	1C	BINARY(4)	Current cluster version
32	20	BINARY(4)	Current cluster version modification level

Offset		Type	Field
Dec	Hex		
36	24	BINARY(4)	Potential node version
40	28	BINARY(4)	Potential node version modification level

Field Descriptions

Bytes available. The number of bytes of data available to be returned to the user.

Bytes returned. The number of bytes of data returned to the user.

Cluster name. The name of the cluster. If no cluster exists on the system, *NONE is returned.

Current cluster version. The version at which the nodes in the cluster are actively communicating with each other. This value in conjunction with the potential node version determines what nodes can join in the cluster. This value also determines the cluster's ability to use new functions supported by the node's potential node version. It is set when the cluster is created and can be changed by the "Adjust Cluster Version (QcstAdjustClusterVersion) API" on page 11. If no cluster exists, the field will be set to 0.

Current cluster version modification level. The modification level of the current cluster version. The modification level further identifies the version at which the nodes in the cluster communicate. It is updated when code changes that impact the version are applied to the system. If no cluster exists, the field will be set to 0.

Potential node version. The version at which the node is capable of communicating with the other nodes in the cluster. This is the value associated with the cluster code installed on the node. It will be used to determine if the node can join a cluster.

Potential node version modification level. The modification level of the potential node version. The modification level further identifies the version at which the node is capable of communicating with the other nodes in the cluster. It is updated when code changes that impact the version are applied to the system and will also be used to determine if the node can join a cluster.

Requesting node id. The node id of the requesting cluster node. If no cluster exists or if the cluster has been created but never started on the system, *NONE is returned.

Reserved. This field will contain hexadecimal zeroes.

Error Messages

Messages that are delivered through the error code parameter are listed here.

Message ID	Error Message Text
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V5R1

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Retrieve Cluster Resource Services Information (QcstRetrieveCRSInfo) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Cluster name	Input	Char(10)
4	Format name	Input	Char(8)
5	Error code	I/O	Char(*)

Service Program: QCSTCTL1
Default Public Authority: *USE
Threadsafe: Yes

The Retrieve Cluster Resource Services Information (QcstRetrieveCRSInfo) API retrieves information about the cluster performance and configuration parameters on a requesting node. The requesting node does not need to be active in the cluster to retrieve the information. This API may be called from a cluster resource group exit program.

Authorities and Locks

None

Required Parameter Group

Receiver variable
OUTPUT; CHAR(*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

Length of receiver variable
INPUT; BINARY(4)

The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

Cluster name
INPUT; CHAR(10)

The name of the cluster for which the information is being retrieved.

Format name
INPUT; CHAR(8)

The content and format of the information that is returned. The possible format names are as follows:

"RCRS0100 Format" on page 54	Returns information about the current settings of the cluster performance and configuration parameters. These parameters may be changed using the "Change Cluster Resource Services (QcstChgClusterResourceServices) API" on page 19.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

RCRS0100 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(4)	Reserved
12	C	BINARY(4)	Configuration tuning level
16	10	BINARY(8)	Receive/Send heartbeat timer ratio
24	18	BINARY(8)	Maximum retry timer ratio
32	20	BINARY(8)	Send heartbeat interval
40	28	BINARY(8)	Retry timer value
48	30	BINARY(8)	CDAT protocol timeout interval
56	38	BINARY(8)	Cluster recovery interval
64	40	BINARY(8)	Maximum retry time
72	48	BINARY(8)	Message fragment size
80	50	BINARY(8)	Send queue overflow
88	58	BINARY(8)	Number of bad messages threshold
96	60	BINARY(8)	Number of ack messages threshold
104	68	BINARY(8)	Unreachable heartbeat ack threshold
112	70	BINARY(8)	Reachable heartbeat ack threshold
120	78	BINARY(8)	Unreachable heartbeat threshold
128	80	BINARY(8)	Reachable heartbeat threshold
136	88	BINARY(8)	Delayed ack timer
144	90	BINARY(8)	Message send window
152	98	BINARY(8)	Enable multicast
160	A0	BINARY(8)	Performance class
168	A8	BINARY(8)	Ack remote fragments

Field Descriptions

Note: Units and ranges for the fields described here may be found in the "Field Settings Range" on page 24 located at the end of the Field Descriptions section of the Change Cluster Resource Services API.

Ack remote fragments. Provides a switch to enable or disable a cluster messaging level acknowledgment for receipt or each fragment sent to a remote cluster node. Fragments are sent by the cluster messaging service for each cluster message whose size is greater than the specified message fragment size. Remote cluster nodes are defined to be any nodes not on the local LAN (having a network or subnet IP address other than that of the source node for the message). ACKing remote fragments may be desirable in those few cases where low bandwidth gateways, routers, or bridges exist between local and remote systems. The valid values for this field are:

0 Acknowledgments are disabled.
1 Acknowledgments are enabled.

Bytes available. The number of bytes of data available to be returned to the user.

Bytes returned. The number of bytes of data returned to the user.

CDAT protocol timeout interval. The timeout value used for distributing the Cluster Destination Address Table (CDAT) and synchronizing cluster communications when doing a create cluster, add node, or start node process. As the number of nodes in the cluster increases, the time required to run this synchronizing protocol increases. This is a low level Cluster Resource Services start-up protocol.

Cluster recovery interval. The interval at which a cluster node takes inventory of required recovery actions and attempts automatic recovery as necessary. Those items checked are:

- Unreachable alternate point-point IP addresses for remote nodes.
- Unreachable multicast IP address for the local subnet.
- Partitioned nodes.

Configuration tuning level. Retrieves the cluster performance and configuration parameters settings. The individual parameter settings for a fast path set option are defined in the "Field Settings Range" on page 24 found in the Change Cluster Resource Service API documentation. The valid values for this field are:

0 Settings have been adjusted individually and are not currently set to one of the fast path settings.

1 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 Normal default values are used for cluster communications performance and configuration parameters.

3 Adjustments are made to cluster communications to » increase « the heartbeating » frequency « and decrease the various message timeout values. With more frequent heartbeats and shorter timeout values, the cluster will be quicker to respond (more sensitive) to communications failures.

Delayed ack timer. The timer used over inbound reliable messages to force an acknowledgment for unacknowledged messages should the sender not have requested an acknowledgment over the last delayed ack time period. This timer is started on receipt of a reliable message and stopped when an acknowledgment is sent for one or more unacknowledged messages.

Enable multicast. The cluster communications infrastructure makes use of User Datagram Protocol (UDP) multicast capabilities as the preferred protocol for sending cluster management information between nodes in a cluster. Where multicast capabilities are supported by the underlying physical media, cluster communications will utilize the UDP multicast to send management messaging from a given node to all local cluster nodes supporting the same subnet address. Messages being sent to nodes on remote networks will always be sent using UDP point to point capabilities. Cluster communications does not rely on routing capability of multicast messages.

The multicast traffic supporting cluster management messaging tends by nature to be bursty. Depending on the number of nodes on a given LAN (supporting 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/second. Bursts of this nature could have a negative impact on older networking equipment. One example would be congestion problems on devices on the LAN serving as Simple Network Management Protocol (SNMP) agents which need to evaluate each and every UDP multicast packet. Some of the earlier networking equipment does not have adequate bandwidth to keep up with this type of traffic. Insure that the network administrator has reviewed the capacity of the networks to handle UDP multicast traffic to make certain that clustering will not have a negative impact on the health and performance of the networks over which it is chosen to operate.

If the network does not wish to have the more efficient multicast capabilities used, setting this field to FALSE (0) will disable the multicast capabilities of the cluster and only point to point communications will be used by the cluster messaging services. The valid values for this field are:

0 Multicast is disabled.

1 Multicast is enabled.

Maximum retry time. Reliable messages are resent at exponentially increasing times should they timeout (that is, not receive a timely acknowledgment). The initial timeout value for a message is the Retry Timer Value and each successive retry builds up by a factor of 2 until the Maximum retry timer value is exceeded. For the default cases, a message would be sent, resent 1 second later, then 2 seconds, 4 seconds, and finally 8 seconds. This represents a total of 15 seconds following which attempts to use alternate IP addressing are tried with the same timer values.

Maximum retry timer ratio. Remote subnets (remote cluster nodes on another LAN/WAN/BUS supporting a different subnet IP address than the sending node) use an extended message timeout value which is based from the Maximum retry time used for local subnets (local cluster nodes supporting the same subnet IP address). For the default case, the Maximum retry time for a local multicast message would be 8 seconds and for a remote point to point message would be 8 x 8 = 64 seconds. This allows for network routing considerations.

Message fragment size. Cluster communications fragments its own messages. This fragment size should be set consistent with the physical media and routing capabilities throughout the network used for clustering. The preferred settings allow for the largest fragment size possible that does not exceed any of the hardware Maximum Transmission Units defined over the entire path so that clustering does all of the fragmentation, not the intermediary networks. The default is set to assume a minimum 1500 byte (less network header space) Ethernet environment.

Message send window. The number of messages allowed outstanding without having received an acknowledgment. The higher the number, the lower the message latency but the larger the required buffer space on a node to save inbound messages.

Number of ack messages threshold. The number of repeat messages that are received over the course of a cluster recovery interval before acknowledgments are sent to multiple source IP addresses for a given node instead of just the current primary address for each message received. While increasing the number of ACKs flowing, this reduces the message resends and latency given that an intermittent communications condition is detected. Eventually, one of the node addresses should be marked as failed and at cluster recovery time, messaging will settle back down using single acknowledgments.

Number of bad messages threshold. The number of undeliverable messages per cluster recovery interval allowed before a failing status is assigned to a node's internet address. At this time, a secondary address (if available) is assigned to be the new primary IP address for the subject remote node.

Performance class. The requested performance characteristics of the cluster communications messaging protocol. Pacing is selectively used for sending out fragments of large messages. Messages are fragmented by the cluster messaging service at the specified message fragment size. The pacing mechanism releases a set number of fragments to the underlying physical layer, then delays, then releases a next set. This is to avoid over running slower physical media. Local here refers to nodes on a local LAN. Remote refers to messaging to cluster nodes on other than the local LAN. Valid values for the performance class are as follows:

0 Normal: Pacing applied to local and remote fragments.
1 High Throughput Local: Pacing applied to remote fragments.
2 High Throughput Local and Remote: No pacing of any fragmented messages.
3 High Throughput Remote: Pacing applied to local fragments.

Reachable heartbeat ack threshold. A node becomes reachable (formerly having been marked as unreachable) from a Cluster Communications heartbeating perspective if ″Reachable heartbeat ack threshold″ (or greater) heartbeat message ACKs are received for the last ″Reachable heartbeat threshold″

heartbeat messages sent to a node. For the default case, a node becomes reachable if 3 or more of the last four heartbeats sent to the marked unreachable node are now acknowledged.

Reachable heartbeat threshold. See Reachable heartbeat ack threshold field description.

Receive/Send heartbeat timer ratio. Ratio of incoming heartbeat messages expected from a neighboring node to the number of heartbeat messages that are sent out. The send rate is always set higher to insure a neighboring node's receive heartbeat timer does not fire under normal operational circumstances.

Reserved. This field will contain hexadecimal zeroes.

Retry timer value. See Maximum retry time field description.

Send heartbeat interval. The interval at which a low level Cluster Communications heartbeat message is sent to a neighboring node.

Send queue overflow. The maximum number of messages that are allowed to be queued up in a Cluster Communications outbound message queue. The CC send queues are distributed amongst the various Distributed Activity (DA) groups. The larger the number, the greater the memory resources that are required to support cluster messaging. If a send queue overflow is hit for a given DA, the inability to send a message could lead to the termination of that DA resulting from the lack of resources on a node.

Unreachable heartbeat ack threshold. A reachable node becomes unreachable from a Cluster Communications heartbeating perspective if ″Unreachable heartbeat ack threshold″ heartbeat message ACKs (or less) are received for the last ″Unreachable heartbeat threshold″ heartbeat messages sent to a node. For the default case, a node becomes unreachable if one or less of the last four heartbeats sent to the marked reachable node are acknowledged.

Unreachable heartbeat threshold. See Unreachable heartbeat ack threshold field description.

Error Messages

Messages that are delivered through the error code parameter are listed here.

Message ID	Error Message Text
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB70 E	API request &1 not compatible with current cluster version.

API introduced: V5R1

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Start Cluster Node (QcstStartClusterNode) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Node entry	Input	Char(*)
4	Format name	Input	Char(8)

5	Results information	Input	Char(30)
6	Error code	I/O	Char(*)

Service Program: QCSTCTL
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Start Cluster Node (QcstStartClusterNode) API is used to start Cluster Resource Services on a node in the cluster. If Cluster Resource Services is successfully started on the node specified, the status of the node will be set to Active. Beginning with cluster version 3, a node can start itself and will be able to rejoin the current active cluster, provided it can find an active node in the cluster.

When a node starts itself, instead of assuming itself as the only active member in the cluster, it will first attempt to find a sponsor node from its cluster membership list. Any active member in the cluster can be a sponsor node. If the joiner finds a sponsor node, the start request will be forwarded to the sponsor node. The joiner then will be started by the cluster (the sponsor node), as if the start request was originated from the sponsor node initially. The sponsor, along with other actives members in the cluster, will process the start-joiner request. The joiner eventually rejoins the cluster.

During the activation of Cluster Resource Services, the allow add to cluster (ALWADDCLU) network attribute is checked to see whether the node being started should be part of the cluster and whether to validate the cluster request through the use of X.509 digital certificates. If validation is required, the requesting node and the node being added » must have the software required for SSL installed on the systems. «

The following conditions apply to this API:

- The node being started must exist in the cluster membership list.
- If all nodes have a status of New, this API must be called from a program running on the node on which the cluster was originally created.
- The node to be started must be IP reachable (TCP/IP is active and the INETD server is started).
- The first time a node is started, this API must be called from a program running on a node that is ACTIVE.
- If all nodes in the cluster are not ACTIVE, this API can be called from a program running on any node that had been previously ACTIVE.
- For cluster version 2 or lower, if any node that had been previously but not currently ACTIVE starts itself, it will be started as a singleton cluster. Starting in cluster version 3, a node which starts itself will be able to rejoin the current active cluster if it can find another active node in the cluster, otherwise it will become a singleton cluster.
- If the cluster is partitioned, this API may be used to start nodes in the partition running the API.
- The potential node version of the node being started must be equal to the current cluster version or up to one level higher than the current cluster version. The potential node version and the current cluster version can be retrieved for all nodes in the cluster by using the “List Cluster Information (QcstListClusterInfo) API” on page 37. The potential node version for the local node can also be retrieved by using the “Retrieve Cluster Information (QcstRetrieveClusterInfo) API” on page 50 or the List Cluster Information API.

If the node being started is in a device domain, this API requires that i5/OS option 41, HA Switchable Resources, is installed and a valid license key exists on that node.

This API operates in an asynchronous mode. See “Behavior of Cluster Resource Services APIs” on page 221 for more information.

Restriction: This API cannot be used in a cluster resource group exit program.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

User Queue Authority
*OBJOPR and *ADD

User Queue Library Authority
*EXECUTE

User Queue Lock
*EXCLRD

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster to which the node belongs.

Node entry
INPUT; CHAR(*)

The node id to be started.

Format name
INPUT; CHAR(8)

The content and format of the information supplied in the node entry array. The possible format names are:

"STRN0100 Format" on page 60	Node id information

Results information
INPUT; CHAR(30)

A library qualified user queue name followed by a reserved field.

Library qualified user queue: A user queue, which exists on the node from which the API was called, that receives results information after the function has completed on all active nodes in the cluster. See the "Usage Notes" on page 60 section of this API for a description of the data that is placed on this queue. This is a 20 character field. The first 10 characters contain the user queue name and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, and *CURLIB are not valid for the library name. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of results information are reserved and must be set to hexadecimal zero.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

STRN0100 Format

Offset		Type	Field
Dec	Hex		
0	0	CHAR(8)	Node id

Field Descriptions

Node id. A unique string of characters that identifies the node to be started.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See "Using Results Information" on page 223 and "Cluster APIs Use of User Queues" on page 221 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Error Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF18BA D	Error occurred with subsystem.
CPF2204 D	User profile &1 not found.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPFBB01 D	Cluster already exists.
CPFBB05 D	Cluster node &1 cannot be started.
CPFBB09 D	Cluster node &1 does not exist in cluster &2.
CPFBB10 D	Specified cluster interface not defined on this system.
CPFBB12 D	Cluster node &1 in cluster &2 could not be started.
CPFBB19 D	Cluster node &1 in cluster &2 already started.
CPFBB24 D	Node &1 not participating in &2 protocol.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB46 D	Cluster Resource Services internal error.
CPFBB54 D	Node &1 not be added to the cluster &2.
CPFBB71 D	Potential node version &1 of node &2 not compatible.
CPFBB79 D	Cluster node &1 could not be started.
CPFBB7A D	Internal Cluster Resource Services mismatch.
CPFBB93 D	Base operating system option 41 not installed or license key not valid.
CPFBB96 D	Internal device domain mismatch.
CPIBB05 I	Cluster node &1 started in cluster &2.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the "Usage Notes" above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3CF1 E	Error code parameter not valid.

Message ID	Error Message Text
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB09 E	Cluster node &1 does not exist in cluster &2.
CPFBB19 E	Cluster node &1 in cluster &2 already started.
CPFBB32 E	Attributes of user queue &1 in library &1 are not valid.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB44 E	&1 API cannot be used within a cluster resource group exit program.
CPFBB46 E	Cluster Resource Services internal error.
CPFBB98 E	Cluster node &1 cannot be started by cluster node &2.

API introduced: V4R4

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Cluster Resource Group APIs

The information provided here includes:

- "Cluster Resource Group APIs—Introduction"
- "Cluster Resource Group API List" on page 74

Cluster Resource Group APIs—Introduction

The Cluster Resource Group (CRG) function within a cluster is to:

- maintain operationally identical cluster resource groups (CRGs) on every node of the cluster resource group recovery domain.
- call the Cluster Resource Group exit program for most cluster resource group APIs
- coordinate activities performed whenever access points for cluster resource groups are changed from one node to another.

Any cluster resource group API may be called on any node in the cluster. Most cluster resource group APIs have an asynchronous behavior.

The majority of the cluster resource group APIs require that Cluster Resource Services be active. This is necessary to ensure consistency of cluster resource groups across the cluster. Each API indicates whether or not Cluster Resource Services needs to be active for the API to complete successfully.

Cluster Resource Services maintains synchronous copies of cluster resource groups (perceptively and operationally identical) on all nodes in the group's recovery domain. When a node joins the cluster or when a cluster partition is resolved, the cluster resource group object is reconciled. This may mean copying the cluster resource group object from some node already in the cluster to the joining node or from the primary partition to nodes in the secondary partition. See "Partition Rules" on page 72 for details on primary and secondary partitions.

Types of Cluster Resource Groups

There are two models of cluster resource groups.

- Primary-backup model. All cluster resource groups of this model define nodes in the recovery domain with a specific role: either primary, backup or replicate. The primary and backup nodes are available to be the access point for the cluster resource. However only one node will be the active access point at a given point in time. This node will be the primary node. Replicate nodes are not available to be an access point. A node role can be changed by assigning the replicate node a role of backup. Examples of cluster resource groups of this model are data, device and application.
- Peer model. All cluster resource groups of this model define nodes in the recovery domain with a role of peer or replicate. The peer nodes are available to be the access point for the cluster resource group. All nodes defined as peer will be the access point at the same time when the cluster resource group is started. Replicate nodes are not available to be an access point. This can be changed by assigning the replicate node a role of peer. Example of cluster resource groups of this model are peer cluster resource groups «.

Cluster resource group objects are either data resiliency, application resiliency, device resiliency» or peer resiliency.« Data resiliency represents multiple copies of data maintained on more than one node in a cluster. Application resiliency enables an application (program) to be restarted on either the same or a different node in the cluster. This is made possible by a Takeover IP Address. Device resiliency allows devices such as auxiliary storage pools to be switched from one node in a cluster to another node.» Peer resiliency represents resources being accessed by mutliple clients.

«

.

Recovery Domain

Cluster resource groups contain a recovery domain. A **Recovery Domain** is that set of cluster nodes which, for a particular cluster resource group, describes the access points of the cluster resource. Each node in the recovery domain is assigned a role that reflects its point of access:

Primary Node The cluster node that is the point of access for the resilient cluster resource. For a replicated resource, the primary node also contains the principal copy of a resource. If this node fails, all cluster resource group objects having this node as the primary access point will failover to a backup node. » This role is allowed for primary-backup model cluster resource groups. «

Backup Nodes Cluster nodes that will take over the role of primary access if the present primary node fails. For a replicated cluster resource, this cluster node contains a copy of that resource. In the case of a data cluster resource group, copies of the data are kept current with replication. » This role is allowed for primary-backup model cluster resource groups. «

Replicate Nodes Cluster nodes that has copies of cluster resources. » The node it is unable to assume the role of primary, backup, or peer «.

» Peer Nodes All nodes have the same copy of the cluster resources. A node defined with this role is available to be the active point of access for the cluster resources. This role is only supported for peer cluster resource groups. «

Some “Cluster Control APIs” on page 1 cause cluster resource group actions to be taken. For example, an “End Cluster Node (QcstEndClusterNode) API” on page 33 will cause the active cluster resource groups on that node to be ended and the “Cluster Resource Group Exit Program” on page 185 to be called. In these instances, the success indicator returned by the exit program will be ignored. The operations will always be considered successful.

A cluster resource group has a recovery domain of one or more cluster nodes. Each cluster node within the recovery domain has two roles: preferred and current. The two node roles need not be the same. When a cluster resource group is initially created, the preferred and the current roles are the same. When a cluster resource group is created, a cluster resource group job is started on each active node in the cluster and a *CRG object will be created on each recovery domain node.

The current role of a node in the recovery domain is changed as a result of operations occurring within the cluster (for example nodes ending, nodes starting, and nodes failing).

» For primary-backup model cluster resource groups «

- When the recovery domain is obtained by the "List Cluster Resource Group Information (QcstListClusterResourceGroupIn) API" on page 141 or when it is passed to the exit program, it is always presented as an array with the primary node first, followed by backup nodes, and finally replicate nodes. If the cluster resource group is active, backup nodes in the recovery domain are ordered so that active nodes appear before nodes that are inactive or partitioned. APIs or cluster events that affect a node's membership status in the recovery domain also cause the order of the backup nodes to change for an active cluster resource group.
- If a cluster resource group is not active, APIs can cause changes to the order of the recovery domain but cluster events such as nodes failing or rejoining the cluster do not. This is done to keep the current recovery domain in the order last requested by the user or the last order when the cluster resource group was ended during times when node failures or rejoins are not important. However, when the cluster resource group becomes active such as with the "Start Cluster Resource Group (QcstStartClusterResourceGroup) API" on page 164, the recovery domain will be reordered if necessary to put active backup nodes before inactive or partitioned backup nodes.
- When the "Initiate Switchover (QcstInitiateSwitchOver) API" on page 135 is used on an active cluster resource group, the first active backup node becomes the new primary and the old primary becomes the last active backup node.
- When the primary node fails for an active cluster resource group, the first active backup node becomes the new primary and the old primary becomes the last inactive backup.

» For peer model cluster resource groups:

- When the recovery domain is obtained by the "List Cluster Resource Group Information (QcstListClusterResourceGroupIn) API" on page 141 or when it is passed to the exit program, it is always presented as an array with the peer nodes first followed by replicate nodes.
- The "Initiate Switchover (QcstInitiateSwitchOver) API" on page 135 is not allowed.
- The nodes are not reordered when nodes fail for an active cluster resource group. «

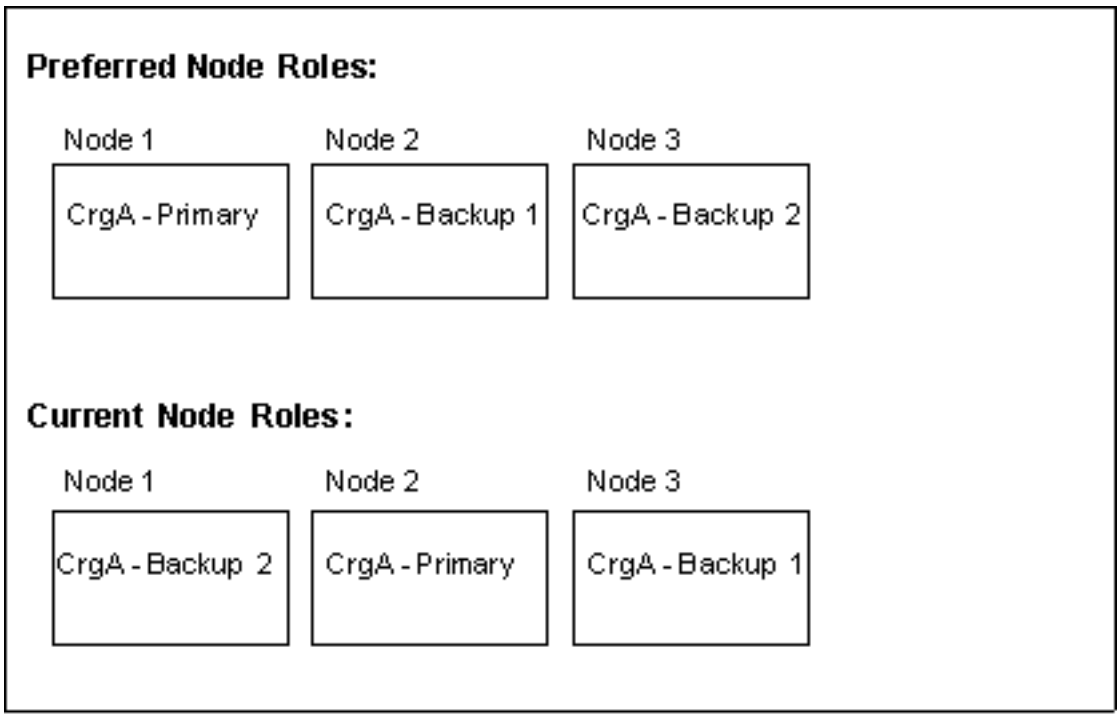
The preferred role of a node in the cluster is changed only by running the following APIs:

- Add Node to Recovery Domain
- Remove Node from Recovery Domain
- Change Cluster Resource Group

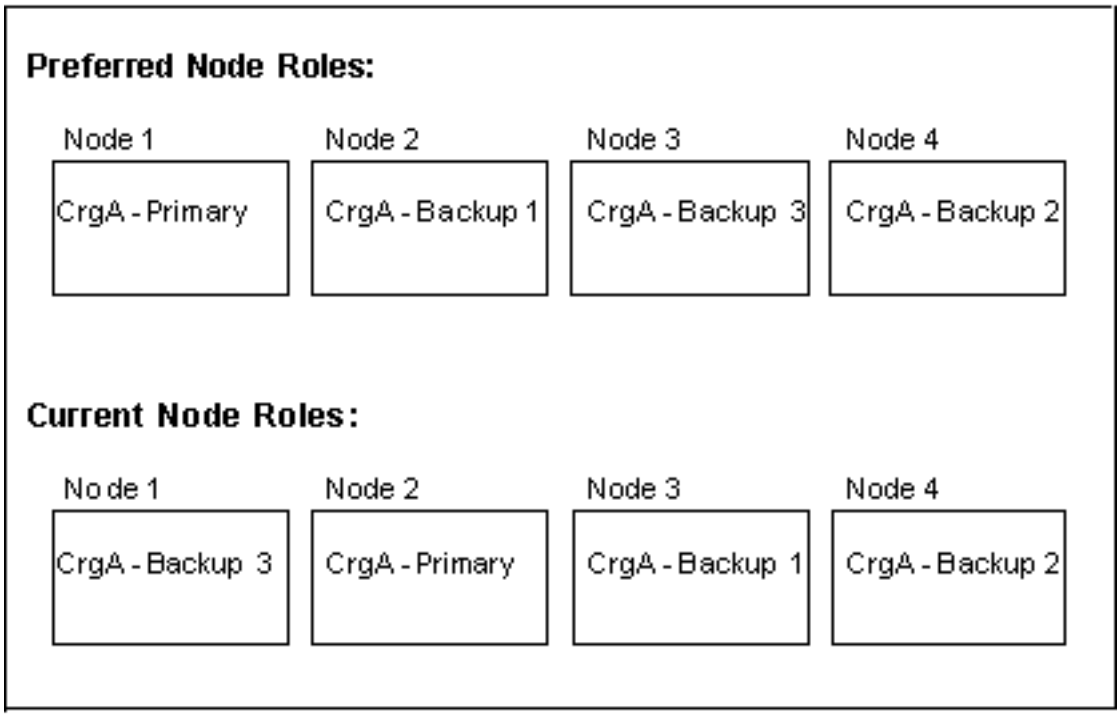
Changes to the node roles are done independently. The role specified for a node in any of these APIs will be assigned to both the current and preferred roles of the node.

Example of node roles for a » primary-backup model cluster resource group. « .

For example, the recovery domain of a » primary-backup model « cluster resource group object has preferred roles of N1-primary, N2-backup1, and N3-backup2, but the current roles are N1-backup2, N2-primary, and N3-backup1. N4 is being added as backup2. Therefore, the preferred roles of the nodes are N1-primary, N2-backup1, N3-backup3, and N4-backup2, and the current roles are N1-backup3, N2-primary, N3-backup1, and N4-backup2.



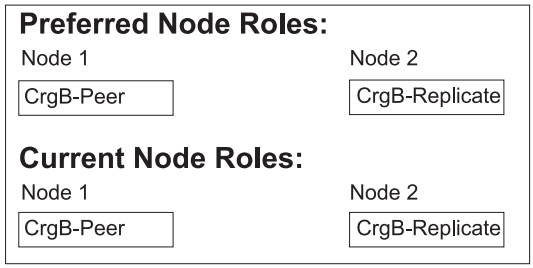
Node 4 being added as Backup 2



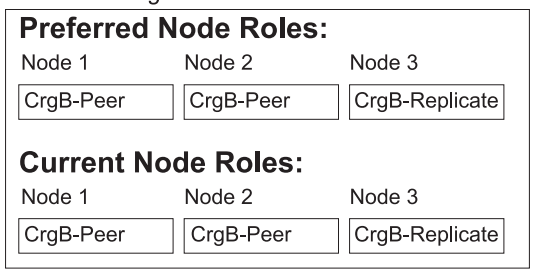
»

Example of node roles for a » peer model cluster resource group.

In this example, the recovery domain of a » peer model cluster resource group object preferred roles are N1-peer, and N2-replicate, but the current roles are N1-peer, and N2-replicate. N3 is being added as peer. Therefore, the preferred roles of the nodes are N1-peer, N2-replicate, and N3-peer, and the current roles are N1-peer, N2-replicate, and N3-peer. Notice that the recovery domain is reordered when another node is added to the recovery domain with a role of peer.



Node 3 being added as a Peer node



RBAFX670-0

«

Exit Program

Every data or application cluster resource group has an associated exit program. A device cluster resource group can also have an exit program but one is not required. This exit program will be called for each of the different action codes listed under the Cluster Resource Group exit program. The exit program is called from a separate job using the user profile supplied when the cluster resource group is created. See “Cluster Resource Group Exit Program” on page 185 for a description of the conditions that cause the exit program to be called.

The user exit program will be restricted from calling some of the APIs. Each API specifies the user exit program restrictions.

Application Takeover IP Address

An » application « takeover IP address is a high availability mechanism used to insulate clients from application server outages. The concept is to use IP address aliasing (multihoming) to define a “floating IP address” associated with multiple application servers or hosts. When one application server in a cluster fails, another cluster node can assume the responsibilities of the application server without requiring the user to reconfigure the clients.

To support address aliasing, application groups contain an IP address resource and a recovery domain. When the application or the node running the application fails, Cluster Resource Services initiates a failover of the group using the IP address to the node assigned the current role of first backup.

The address specified for the takeover IP address must not be used for any other purposes. Cluster Resource Services will not allow certain API operations to complete successfully if the IP address is in use. This restriction ensures that the structures being created will provide application resilience.

Server Takeover IP Address

A server takeover IP address is just like the » application « takeover IP address for an application CRG, except it is used for servers associated with the relational database name in the device description for an auxiliary storage pool. The address can only be specified for a primary auxiliary storage pool. Only one IP address can be specified per primary auxiliary storage pool. The address must be unique, and must not be used for any other purpose.

The user is responsible for configuring and managing the server takeover IP address. The IP address must be added on all nodes in the recovery domain prior to starting the cluster resource group. Starting of a device cluster resource group will not start the server IP address or vary on the device. That is the user's responsibility. Cluster Resource Service manages the IP address only during a switchover or failover.

On switchover or failover, clustering will end the IP address on the current primary, and uses the value in the "configuration object online" field to determine what action should occur on the new primary node. Based on the value in the "configuration object online" field it will either start the IP address and vary on the device or do nothing to the IP address and device.

Failover Message Queue

A failover message queue allows a user to control what happens at failover time. A failover policy could be:

- failover continues to act like it did for V5R1M0 and prior.
- failover sends an inquiry message to the failover message queue and waits the specific amount of time specified by the user.

A failover message queue may be specifed when a cluster resource group is created. A message will be placed on the queue when the primary node of the active cluster resource group either ends or fails, forcing the cluster resource group to fail over to a new primary. In the case of a node failure, each cluster resource group will enqueue a separate message to its failover message queue if one is defined. No message will be enqueued if the primary node is removed from the cluster.

The message will be placed on the message queue on the new primary node before the call to the exit program. This gives the user the option of continuing the failover to the new primary, or cancelling the failover. If the failover is cancelled, the primary node will not be changed, and the cluster resource group will become Inactive. The exit program will be called with an action code of Failover Cancelled.

There are two associated parameters with the qualified failover message queue. The failover wait time allows the user to specify how long Cluster Resource Services should wait for a reply to the failover message. The user can choose to wait forever, proceed with failover without waiting for a reply, or wait a specified number of minutes. The failover default action allows the user to choose whether to continue or cancel failover if a reply to the failover message is not received within the time limit specified in the failover wait time parameter or if the message cannot be enqueued for some reason.

Site Name and Data Port IP Addresses

Site name and data port IP addresses are associated with a recovery domain node for a device CRG, applicable only to cross-site mirroring. Both must be specified together for a recovery domain node. That is, a node which has a site name must also have at least one data port IP address specified.

Geographic mirroring, which is a subfunction of cross-site mirroring, supports two physical copies of auxiliary storage pool, one on each site. Only two sites are supported. A site primary node is the node which has the highest node role ranking for that site. A production site primary node, which is also the primary node for a device CRG, owns a production copy of the auxiliary storage pool. A mirror site primary node, which is the backup node which has the highest node role ranking at the mirror site, owns a mirror copy of the auxiliary storage pool.

A site may contain one or more recovery domain nodes at the same physical location. All nodes at a site must have access to the same physical copy of auxiliary storage pool. If there is only one node at a site, the auxiliary storage pool on that site does not need to be switchable. A node which belongs to more than one device CRG may or may not have the same site name.

Geographic mirroring is performed by sending updates from a production site primary node to a mirror site primary node via data port IP addresses. Each recovery domain node could have up to four data port IP addresses. They must be unique across all recovery domain nodes and CRGs.

User is responsible for configuring and managing data port IP addresses. They must already exist on all nodes in the recovery domain prior to starting a device CRG. Clustering will not start or end data port IP addresses under any circumstances, including starting and ending of a cluster resource group, switchover and failover. User must start the data port IP addresses before geographic mirroring can be performed. It is recommended that data port IP addresses are dedicated for geographic mirroring use only. It is also recommended that multiple data port IP addresses on each recovery domain node map to different adapters. This will help to avoid a single point of failure on the adapter and also improve performance of geographic mirroring.

Summary of Cluster Resource Group Status

Each cluster resource group has a status associated with it. The status of the cluster resource group may govern the behavior of a particular API call. In the following list of values, an indication of what happens when the exit program completes successfully applies only to a cluster resource group which has an exit program. If no exit program was specified, the same action occurs as for a successful completion. The possible values are:

10 **Active.** The resources managed by the cluster resource group are currently resilient.

20 **Inactive.** The resources managed by the cluster resource group are currently not resilient.

30 **Indoubt**. The information contained within the cluster resource group object may not be accurate. This status occurs when an exit program is called with an action of Undo and fails to complete successfully.

40 **Restored.** The cluster resource group object was restored on this node and has not been copied to the other nodes in the recovery domain. When Cluster Resource Services is started on this node, the cluster resource group will be synchronized with the other nodes in the recovery domain and its status set to Inactive.

500 **Add Node Pending.** A new node is in the process of being added to the recovery domain of a cluster resource group. If the exit program is successful the status is reset to its value at the time the API was called. If the exit program fails and the original state cannot be recovered, the status is set to Indoubt.

510 **Delete Pending.** Cluster resource group object is in the process of being deleted. When the exit program completes the cluster resource group is deleted from all nodes in the recovery domain.

520 **Change Pending.** The cluster resource group is in the process of being changed. If the exit program is successful the status is reset to the value at the time the API was called. If the exit program fails and the original state cannot be recovered, status is set to Indoubt.

530 **End Cluster Resource Group Pending.** Resiliency for the cluster resource group is in the process of ending. If the exit program is successful, the status is set to Inactive. If the exit program fails and the original state cannot be recovered, the status is set to Indoubt.

540 **Initialize Pending.** A cluster resource group is being created and is in the process of being initialized. If the exit program is successful, the status is set to Inactive. If the exit program fails, the cluster resource group will be deleted from all nodes.

550 **Remove Node Pending.** A node is in the process of being removed from the recovery domain of the cluster resource group. If the exit program is successful, the status is reset to the value at the time the API was called. If the exit program fails and the original state cannot be recovered, the status is set to Indoubt.

560 **Start Cluster Resource Group Pending.** Resiliency is in the process of starting for the cluster resource group. If the exit program is successful, the status is set to Active. If the exit program fails and the original state cannot be recovered, the status is set to Indoubt. For peer model cluster resource groups all nodes defined with a role of peer are active access points for the cluster resources.

570 **Switchover Pending.** The Initiate Switchover API was called, a failure of a cluster resource group occurred, or a node failed, causing a switchover or failover to begin. The first backup node is in the process of becoming the primary node. If the exit program is successful, the status is set to Active. If the exit program fails and the original state cannot be recovered, the status is set to Indoubt. While the switchover function is not valid for a peer cluster resource group, users may see the status ″switchover pending″ during a node failure.

580 **Delete Command Pending.** Cluster resource group object is being deleted by the Delete Cluster Resource Group (DLTCRG) command. The Cluster resource group object is only removed from the node running the command. This is not a distributed request. At the completion of the command, the cluster resource group is deleted from the node.

590 **Add Device Entry Pending.** A device entry is being added to a cluster resource group. If the exit program is successful, the status is reset to its value at the time the API was called. If the exit program fails and the original state cannot be recovered, the status is set to Indoubt.

600 **Remove Device Entry Pending.** A device entry is being removed from a cluster resource group. If the exit program is successful, the status is reset to its value at the time the API was called. If the exit program fails and the original state cannot be recovered, the status is set to Indoubt.

610 **Change Device Entry Pending.** A device entry is being changed in a cluster resource group. If the exit program is successful, the status is reset to its value at the time the API was called. If the exit program fails and the original state cannot be recovered, the status is set to Indoubt.

620 **Change Node Status Pending.** The status of a node in the cluster resource group″s current recovery domain is being changed. If the change is successful, the status is reset to its value at the time the Change Cluster Node Entry API was called. Failure of the exit program causes the status of the cluster resource group to be set to Indoubt. If a backup node is reassigned as the primary node for a resilient device cluster resource group and the ownership of the device cannot be transferred to the new primary node, the status is set to Indoubt.

The relationship between the cluster resource group status and the cluster resource group APIs is summarized in the following table. See the cluster resource group APIs for additional details on the cluster resource group status.

Summary of cluster resource group statuses for affected Cluster Resource Services API

Cluster Resource Services API	Original status	Status while exit program running	Action Code	Status - exit program successful	Status - exit program failure on Undo
Add Cluster Resource Group Device Entry	• Active • Inactive • Indoubt • Restored - ERROR • Any pending status - ERROR	Add Device Entry Pending	17 - Add Device Entry	original status	Indoubt
Add Node to Recovery Domain	• Active – Adding primary - Error – Adding backup, replicate » or peer « • Inactive • Indoubt • Restored - ERROR • Any pending status - ERROR	Add Node Pending	11 - Add Node	original status	Indoubt
Change Cluster Node Entry	When changing node status: • Active • Inactive • Indoubt • Restored - ERROR • Any pending status - ERROR	Change Node Status Pending	20 - Change Node Status	original status	Indoubt ** Indoubt if device ownership cannot be transferred for a resilient device cluster resource group.

Cluster Resource Services API	Original status	Status while exit program running	Action Code	Status - exit program successful	Status - exit program failure on Undo
Change Cluster Resource Group	If changing node to primary or changing takeover IP address: • Active — ERROR • Inactive • Indoubt • Restored - ERROR • Any pending status - ERROR All other changes: • Active • Inactive • Indoubt • Restored - ERROR • Any pending status - ERROR	Change Pending	13 - Change **Note:** Only call exit program for changing node role in recovery domain.	original status	Indoubt
Change Cluster Resource Group Device Entry	• Active • Inactive • Indoubt • Restored - ERROR • Any pending status - ERROR	Change Device Entry Pending	19 - Change Device Entry	original status	Indoubt
Create Cluster Resource Group	N/A	Initialize Pending	1 - Initialize	Inactive	*CRG deleted
Delete Cluster Resource Group	• Active — ERROR • Inactive • Indoubt • Restored • Any pending status - ERROR	Delete Pending	• 5 - Verification Phase • 7 - Delete	*CRG deleted	• orginal status (if during Verification Phase, no undo, *CRG not deleted) • *CRG deleted (if during Delete) ** Indoubt if Cluster Resource Services fails

Cluster Resource Services API	Original status	Status while exit program running	Action Code	Status - exit program successful	Status - exit program failure on Undo
End Cluster Resource Group	• Active • Inactive — ERROR • Indoubt • Restored — ERROR • Any pending status - ERROR	End Cluster Resource Group Pending	4 - End	Inactive	Indoubt
End Cluster Node	• Active • Inactive • Indoubt • Restored • Any pending status	Switchover Pending	• 16 - End Node for the node ending • 9 - Failover for other nodes	original status	Indoubt
Initiate Switchover	• Active • Inactive — ERROR • Indoubt — ERROR • Restored — ERROR • Any pending status - ERROR • » Any peer cluster resource group - ERROR «	Switchover Pending	10 - Switchover **Note:** If application cluster resource group, exit program called again with action Start.	Active	Indoubt
Remove Cluster Node Entry	• Active • Inactive • Indoubt • Restored • Any pending status	Switchover Pending	• 12 - Remove Node for the node being removed • 9 - Failover for other nodes	original status	Indoubt
Remove Cluster Resource Group Device Entry	• Active - ERROR if last device entry removed • Inactive • Indoubt • Restored - ERROR • Any pending status - ERROR	Remove Device Entry Pending	18 - Remove Device Entry	original status	Indoubt

Cluster Resource Services API	Original status	Status while exit program running	Action Code	Status - exit program successful	Status - exit program failure on Undo
Remove Node From Recovery Domain	• Active – Removing primary - ERROR – Removing backup, replicate » or peer « • Inactive • Indoubt • Restored — ERROR • Any pending status - ERROR	Remove Node Pending	12 - Remove Node	original status	Indoubt
Start Cluster Node Entry	• Active • Inactive • Indoubt • Restored • Any pending status	» No pending value used «	8 - Rejoin	original status	Indoubt
Start Cluster Resource Group	• Active — ERROR • Inactive • Indoubt • Restored — ERROR • Any pending status - ERROR	Start Cluster Resource Group Pending	2 - Start	Active	Indoubt

Partition Rules

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 cluster resource groups: «

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.

» For peer model cluster resource groups:

- 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 partion. 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.
- 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 API when in a partition state are:

Add Cluster Resource Group Device Entry
Allowed only in a primary partition and all nodes in the cluster resource group″s recovery domain must be active in the primary partition.

Add Node to Recovery Domain
Allowed only in a primary partition.

Change Cluster Resource Group
Allowed only in a primary partition.

Change Cluster Resource Group Device Entry
Allowed only in a primary partition.

Create Cluster Resource Group
Not allowed in any partition.

Delete Cluster Restore Group
Allowed in any partition, but only affects partition running the API.

Distribute Information
Allowed in any partition, but only affects partition running the API.

End Cluster Resource Group
Allowed only in a primary partition. » Allowed in all partitions for peer cluster resource groups, but only affects the partition running the API. «

Initiate Switchover
Allowed only in a primary partition.

List Cluster Resource Groups
Allowed in any partition.

List Cluster Resource Group Information
Allowed in any partition.

Remove Cluster Resource Group Device Entry
Allowed only in a primary partition.

Remove Node from Recovery Domain
Allowed only in a primary partition.

Start Cluster Resource Group.
Allowed only in a primary partition. » 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 resynchronized 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 will be copied to nodes in a secondary partition. » When merging two secondary partitions for peer-model, the partition which has cluster resource group with status of Active will override the other partition. If both partitions have the same status for cluster resource group, the partition which contains the first active node listed in the cluster resource group recovery domain will be copied to all nodes in the recovery domain. The version of the cluster resource group in the winning partition will be copied to nodes in the overridden partition. «

On occasion, a partition condition may be reported incorrectly and one or more nodes may have actually failed. If one of these failed nodes has the current role of primary for a cluster resource group, special recovery actions are required in order to assign the primary node role to a node in a secondary partition.

After these actions have been taken, returning the failed nodes to the cluster becomes much more difficult. Thus, these actions should be taken only when the failed node will be unavailable for an extended period of time. An example of when to do this would be the loss of a primary site.

The Change Cluster Node Entry API may be used to tell Cluster Resource Services that a node has really failed rather than partitioned. Once all nodes have been identified as failing, the List Cluster Resource Group Information API can be used to determine if the recovery domain has been reordered as the situation requires, and the Start Cluster Resource Group API can be used to restart the cluster resource group.

See "Change Cluster Node Entry (QcstChangeClusterNodeEntry) API" on page 14 for additional information.

Cluster Resource Group API List

The cluster resource group APIs are:

- "Add Cluster Resource Group Device Entry (QcstAddClusterResourceGroupDev) API" on page 75 (QcstAddClusterResourceGroupDev) adds a new device entry to a cluster resource group.
- "Add Node To Recovery Domain (QcstAddNodeToRcvyDomain) API" on page 81 (QcstAddNodeToRcvyDomain) adds a new node to the recovery domain of an existing cluster resource group.
- "Change Cluster Resource Group (QcstChangeClusterResourceGroup) API" on page 86 (QcstChangeClusterResourceGroup) changes some of the attributes of a cluster resource group.
- "Change Cluster Resource Group Device Entry (QcstChgClusterResourceGroupDev) API" on page 102 (QcstChgClusterResourceGroupDev) changes a device entry in a cluster resource group.
- "Create Cluster Resource Group (QcstCreateClusterResourceGroup) API" on page 107 (QcstCreateClusterResourceGroup) creates a cluster resource group object.
- "Delete Cluster Resource Group (QcstDeleteClusterResourceGroup) API" on page 124 (QcstDeleteClusterResourceGroup) deletes a cluster resource group.
- "Distribute Information (QcstDistributeInformation) API" on page 127 (QcstDistributeInformation) delivers information from a node in the recovery domain to other nodes in the recovery domain.
- "End Cluster Resource Group (QcstEndClusterResourceGroup) API" on page 132 (QcstEndClusterResourceGroup) calls the Cluster Resource Group Exit Program to disable the resilience of the objects or application.
- "Initiate Switchover (QcstInitiateSwitchOver) API" on page 135 (QcstInitiateSwitchOver) changes the current recovery domain of a cluster resource group by making the primary node the last backup node and first backup node the primary node.
- "List Cluster Resource Group Information (QcstListClusterResourceGroupIn) API" on page 141 (QcstListClusterResourceGroupIn) returns the contents of a cluster resource group object.
- "List Cluster Resource Groups (QcstListClusterResourceGroups) API" on page 152 (QcstListClusterResourceGroups) generates a list of cluster resource groups and descriptive information about them.
- "Remove Cluster Resource Group Device Entry (QcstRmvClusterResourceGroupDev) API" on page 156 (QcstRmvClusterResourceGroupDev) removes a device entry from a cluster resource group.
- "Remove Node From Recovery Domain (QcstRemoveNodeFromRcvyDomain) API" on page 160 (QcstRemoveNodeFromRcvyDomain) removes a node from the recovery domain of an existing cluster resource group.
- "Start Cluster Resource Group (QcstStartClusterResourceGroup) API" on page 164 (QcstStartClusterResourceGroup) calls the Cluster Resource Group Exit Program to enable resilience for the objects or application.

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Add Cluster Resource Group Device Entry (QcstAddClusterResourceGroupDev) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Cluster resource group name	Input	Char(10)
4	Configuration object entry information	Input	Char(*)
5	Format name	Input	Char(8)
6	Results information	Input	Char(30)
7	Error code	I/O	Char(*)

Service Program: QCSTCRG1
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Add Cluster Resource Group Device Entry (QcstAddClusterResourceGroupDev) API adds one or more configuration objects representing hardware devices to a device cluster resource group. All devices being added must be able to be switched from one cluster node to another, unless for cross-site mirroring where a site only has one node.

If the cluster resource group contains any members of an auxiliary storage pool group, it must contain **all** members before the cluster resource group can be started. All members do not have to be specifed at once. Additional members can be added later. If the auxiliary storage pool group has previously been created and clustering can determine which members are in the group, a warning message is sent if some members of the group were not added.

If an exit program is specified for the cluster resource group, the cluster resource group exit program is called with an action code of Add Device Entry (17) on all active nodes in the recovery domain. The cluster resource group status is set to Add Device Entry Pending (590). If the exit program completes successfully, the cluster resource group status is reset to its value at the time the API was called. If the exit program fails and the cluster resource group cannot be restored to its original condition, the cluster resource group status is set to Indoubt (30).

This API requires:

1. Only auxiliary storage pool devices are supported.
2. Cluster Resource Services must be active on the node processing the request.
3. The number of configuration objects being added plus the number of configuration objects already in the cluster resource group cannot exceed 256.
4. The configuration object for the devices being added must exist on all nodes in the recovery domain of the cluster resource group.
5. The resource name specified in the configuration object must be the same on all nodes in the recovery domain.
6. If a data base name is specified in the configuration object, it must be the same on all nodes in the recovery domain.
7. If a new auxiliary storage pool group is added to an active cluster resource group, all members of the group must be specified.
8. If a server takeover IP address is specified, it must exist on all nodes in the recovery domain if the cluster resource group is active. The server takeover IP address must be unique. It can only be associated with a primary auxiliary storage pool.
9. The configuration objects being added cannot be specified in another cluster resource group.

10. Devices attached to the same IOP or high-speed link I/O bridge can be specified for only one cluster resource group. For cross-site mirroring which only has one node at a site, this requirement does not apply.
11. If devices attached to different IOPs or high-speed link I/O bridges are grouped such as for an auxiliary storage pool, all devices for the affected IOPs or high-speed link I/O bridges must be specified in the same cluster resource group. For cross-site mirroring which only has one node at a site, this requirement does not apply.
12. The IOP or high-speed link I/O bridge controlling the devices specified in a cluster resource group must be accessible by all nodes in the cluster resource group's recovery domain or by all nodes within the same site (for cross-site mirroring). This is verified if sufficient hardware configuration has been performed so that all nodes are aware of the new hardware. If hardware configuration is incomplete, this is verified when the "Start Cluster Resource Group (QcstStartClusterResourceGroup) API" on page 164 is called.
13. If the primary node does not currently own the specified devices, the API fails with an error message.
14. If an exit program is specified, the exit program must exist on all nodes in the recovery domain.
15. All nodes in the recovery domain must be active.

This API operates in an asynchronous mode. See "Behavior of Cluster Resource Services APIs" on page 221 for more information.

Restriction: This API cannot be called from a cluster resource group exit program.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

Cluster Resource Group Authority
*CHANGE

Cluster Resource Group Library Authority
*EXECUTE

Cluster Resource Group Lock
*EXCL

Exit Program Authority » (applies to user profile calling the API and user profile to run the exit program) «
*EXECUTE

Exit Program Library Authority » (applies to user profile calling the API and user profile to run the exit program) «
*EXECUTE

User Profile Authority » (applies to user profile to run the exit program) «
*USE

Request Information User Queue Authority
*OBJOPR, *ADD

Request Information User Queue Library Authority
*EXECUTE

Request Information User Queue Lock
*EXCLRD

Configuration Object Authority
*USE and *OBJMGT

» *Configuration Object Lock*
*EXCLRD «

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster to which the cluster resource group belongs.

Cluster resource group name
INPUT; CHAR(10)

The name of the cluster resource group which is to be changed.

Configuration object entry information
INPUT; CHAR(*)

Detailed information about the configuration objects to be added to the cluster resource group. For more information, see "Device Resiliency (RGDA0100 Format)."

Format name
INPUT; CHAR(8)

The content and format of the device information. The possible format names are:

"Device Resiliency (RGDA0100 Format)"	This format describes the resilient device.

Results information
INPUT; CHAR(30)

This parameter identifies a qualified user queue field and is followed by a reserved field.

Qualified user queue: Completion information is returned to this user queue, which exists on the node from which the API was called, after the function has completed. See the "Usage Notes" on page 79 section of this API for a description of the data that is placed on this queue. This is a 20-character field. The first 10 characters contain the user queue name, and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, *CURLIB are not valid library names. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of the 30-character results information are reserved. Each character in this field must be set to hexadecimal zero.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Device Resiliency (RGDA0100 Format)

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Offset to configuration object array
4	4	BINARY(4)	Number of entries in configuration object array
8	8	BINARY(4)	Length of configuration object array entry
12	C	BINARY(4)	Offset to additional fields

Offset		Type	Field
Dec	Hex		
16	E	BINARY(4)	Length of additional fields
*	*	Array (*) of CHAR(36)	Configuration object array
These fields repeat, in the order listed, for each device.		CHAR(10)	Configuration object name
		CHAR(2)	Reserved
		BINARY(4)	Configuration object type
		BINARY(4)	Configuration object online
		CHAR(16)	Server takeover IP address

Field Descriptions

Configuration object array.

This array identifies the resilient devices.

Configuration object name. The name of the auxiliary storage pool device description object which can be switched between the nodes in the recovery domain. An auxiliary storage pool device description can be specified in only one cluster resource group.

Configuration object online. Vary the configuration object on and start the server takeover IP address or leave the configuration object varied off and the server takeover IP address inactive when a device is switched from one node to another with the "Initiate Switchover (QcstInitiateSwitchOver) API" on page 135 or when it is failed over to a backup node. This attribute does **not** vary the device on or off and does **not** start or end the server takeover IP address when the cluster resource group is started or ended and when adding a new device entry to the cluster resource group. For secondary auxiliary storage pools, only a value of 2 is valid. If cluster resources cannot determine if this value is correct for a device entry because the auxiliary storage pool is not yet created, any errors will be detected when the cluster resource group is started. A value of 2 cannot be specified for any other device type. Possible values are:

0 Do not vary the configuration object on and do not start the server takeover IP address.
1 Vary the configuration object on and start the server takeover IP address.
2 Perform the same action for a secondary auxiliary storage pool as is specified for the primary.

Configuration object type. This specifies the type of configuration object specified with configuration object name. Possible values are:

1 Device description

Length of additional fields. The length in bytes of additional fields. This must be set to hexadecimal zero. It will be used in a future release if more fields are needed in the RGDA0100 format.

Length of configuration object array entry. This specifies the length of an entry in the configuration object array. Is values must be set to the length of an array entry.

Number of entries in configuration object array. The number of entries in the configuration object array. This must be greater than zero and the number of these entries plus the number of entries already in the cluster resource group cannot be greater than 256.

Offset to additional fields. The byte offset from the beginning of this parameter to additional fields. This must be set to hexadecimal zero. It will be used in a future release if more fields are needed in the RGDA0100 format.

Offset to configuration object array. The byte offset from the beginning of this parameter to the configuration object array field.

Reserved. Must contain hexadecimal zeroes.

Server takeover IP address. This is a takeover IP address for servers associated with the relational database name in the device description for an auxiliary storage pool. This field is optional and can only be specified for a primary auxiliary storage pool. If specified, the address must be represented in dotted decimal format and be a null-terminated string. The specified address must exist on all nodes in the recovery domain if the cluster resource group is active. If not specified, or for a secondary and UDFS auxiliary storage pool, this field must be set to *NONE and be left justified. If the current cluster version is 2 and the length of configuration object array entry specified includes the server takeover IP address, this field must be set to hexadecimal zeroes. Valid special values for this field are:

**NONE* There is no server takeover IP address associated with the relational database name in the device description for an auxiliary storage pool.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See "Cluster APIs Use of User Queues" on page 221 and "Using Results Information" on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Error Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF18BA D	Error occurred with subsystem.
CPF2113 E	Cannot allocate library &1.
CPF2204 D	User profile &1 not found.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPF9801 D	Object &2 in library &3 not found.
CPF9802 D	Not authorized to object &2 in &3.
CPF9803 D	Cannot allocate object &2 in library &3.
CPF9804 D	Object &2 in library &3 damaged.
CPF9810 D	Library &1 not found.
CPFBB0B D	Request using takeover IP address &1 failed.
CPFBB0F D	Cluster resource group &1 does not exist in cluster &2.
CPFBB18 D	Request &1 is not allowed for cluster resource group &2.
CPFBB2C D	Attributes of exit program &1 in library &2 are not valid.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB2E D	Job submission failed for cluster resource group &1 in cluster &2.
CPFBB32 D	Attributes of user queue &1 in library &2 are not valid.
CPFBB35 D	The user profile name &1 is not valid for this request.
CPFBB38 D	Library name &1 is not allowed for this request.
CPFBB39 D	The current user does not have IOSYSCFG special authority.
CPFBB46 D	Cluster Resource Services internal error.
CPIBB10 D	Cluster resource group exit program &1 in library &2 on node &3 failed.
CPFBB5B D	Resource name &1 incorrect for configuration object &2 on node &3.

Message ID	Error Message Text
CPFBB5C D	Configuration object &1 already in cluster resource group &2.
CPFBB5D D	Other related devices already in cluster resource group &1.
CPFBB64 D	Configuration object &1 not valid device type.
CPFBB66 D	Request failed for device cluster resource group &3.
CPFBB70 D	API request &1 not compatible with current cluster version.
CPFBB7A D	Primary node &1 in cluster resource group &2 not current owner of specified devices.
CPFBB7B D	Device type incorrect for configuration object &1 on node &2.
CPFBB7C D	Resource name &1 already used by configuration object &2 in cluster resource group &4.
CPFBB7D D	Configuration object &1 already in cluster resource group &2.
CPFBB7E D	Resource name &1 already in cluster resource group &2.
CPFBB7F D	Too many I/O processors or high-speed link I/O bridges specified for cluster resource group &1.
CPFBB80 D	Request failed for device cluster resource group &3.
CPFBB90 D	Request failed for device cluster resource group &3.
CPFBB97 D	Primary node does not own hardware for configuration object &1.
CPFBB98 D	Hardware resource &1 not switchable.
CPFBB99 D	Request failed for device cluster resource group &3.
CPFBB9A D	Online value not valid for device &1.
CPFBB9B D	Auxiliary storage pool group member &1 not specified.
CPFBB9C D	Not all auxiliary storage pool group members added or removed together.
CPFBB9D D	Device &1 not compatible with current cluster version.
CPFBB9E D	Data base name &1 not correct for configuration object &2 on node &3.
CPFBBA6 E	Server takeover IP address cannot be associated with device subtype &1.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the "Usage Notes" on page 79 above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF2204 E	User profile &1 not found.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C29 E	Object name &1 is not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB0F E	Cluster resource group &1 does not exist in cluster &2.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB2C E	Attributes of exit program &1 in library &2 are not valid.
CPFBB32 E	Attributes of user queue &1 in library &2 not valid.
CPFBB35 E	The user profile name &1 is not valid for this request.
CPFBB38 E	Library name &1 is not allowed for this request.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB44 E	&1 API cannot be called from a cluster resource group exit program.

Message ID	Error Message Text
CPFBB5F E	Number of configuration object entries not valid.
CPFBB60 E	Offset to configuration object array is not valid.
CPFBB61 E	Configuration object &1 specified more than once in configuration object array.
CPFBB63 E	The value specified for the field at offset &1 of configuration object array entry &2 is not valid.
CPFBB64 E	Configuration object &1 not valid device type.
CPFBB6B E	Request not valid for type &1 cluster resource group.
CPFBB70 E	API request &1 not compatible with current cluster version.
CPFBB7A E	Primary node &1 in cluster resource group &2 not current owner of specified devices.
CPFBBA5 E	Server takeover IP address &1 specified more than once in the configuration object array.
TCP1901 D	Internet address &1 not valid.

API introduced: V5R1

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Add Node To Recovery Domain (QcstAddNodeToRcvyDomain) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Cluster resource group name	Input	Char(10)
4	Node id	Input	Char(8)
5	Node role	Input	Binary(4)
6	Results information	Input	Char(30)
7	Error code	I/O	Char(*)

Service Program: QCSTCRG1
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Add Node To Recovery Domain API is used to add a new node to the recovery domain of an existing cluster resource group. This API causes the preferred and current roles of all nodes in the recovery domain to be updated.

This API will do the following:

1. Set the cluster resource group status to Add Node Pending(500).
2. For application cluster resource groups:
   a. If Cluster Resource Services configured the takeover IP address, add the interface. If the takeover IP address cannot be added, this request fails.
   b. If the cluster resource group is active and the node is being added as a backup, verify the takeover IP address exists and is not active on the node being added.
3. For device cluster resource groups:
   a. If the node being added to a cluster resource group is to become the new primary node, ownership of the devices specified in the cluster resource group are switched from the current primary to the new primary if none of the » current primary « devices are varied on and the cluster resource group is not active. If any devices are varied on, an error message is returned. Devices are not varied on after the ownership is switched.
4. Call the cluster resource group exit program with the action code of Add Node (11) on all active nodes in the recovery domain, if an exit program is specified for the cluster resource group.
5. Verify the queue used by the "Distribute Information (QcstDistributeInformation) API" on page 127 if the cluster resource group has been created to indicate the Distribute Information API will be used.
6. Verify the failover message queue if one was specified when the cluster resource group was created.

7. Reset the cluster resource group status to the value at the time the API was called if the exit program is successful.
8. Set the cluster resource group status to Indoubt(30) if the exit program fails and the original state of the cluster resource group cannot be recovered.
9. Assign the added node the specified role. » For primary-backup model cluster resource groups, if the cluster resource group is active, it has more than one backup node, and some backup nodes are not active, the recovery domain may be reordered so that all active backup nodes are ordered before inactive backup nodes.«

To change the role of the added node, use the "Change Cluster Resource Group (QcstChangeClusterResourceGroup) API" on page 86.

To remove a node from the recovery domain use the "Remove Node From Recovery Domain (QcstRemoveNodeFromRcvyDomain) API" on page 160.

This API operates in an asynchronous mode. See "Behavior of Cluster Resource Services APIs" on page 221 for more information.

Restrictions:

1. This API cannot be called from a cluster resource group exit program.
2. Cluster Resource Services must be active on the node processing the request.
3. At least one active node in the recovery domain.
4. The node being added must be active in the cluster.
5. The cluster resource group exit program must exist on each node in the recovery domain, including the node being added. The exit program cannot be in an independent auxiliary storage pool.
6. The node being added must not already be a member of the cluster resource group recovery domain.
7. » For primary-backup model cluster resource groups,« a node can only be added as a primary node if the cluster resource group has a status of Inactive (20). The old primary becomes the last backup. If the cluster resource group has a status of Active(10), a node can be added as either a backup or a replicate.
8. » For peer model cluster resource groups, the node can be added as a peer or replicate. If the cluster resource group has a status of Active (10) and a peer node is added, the node will be added as an active access point. «
9. The queue used by the Distribute Information API must exist on each node in the recovery domain, including the node being added if it was specified on the "Create Cluster Resource Group (QcstCreateClusterResourceGroup) API" on page 107. This is verified after the exit program returns. The distributed information user queue does not allow pointers within the message content. The distributed information user queue cannot be in an independent auxiliary storage pool.
10. ». If the failover message queue was defined on the cluster resource group, « the failover message queue must exist on each node in the recovery domain, including the node being added. This is verified after the exit program returns. The queue cannot be in an independent auxiliary storage pool.
11. For device cluster resource groups:
    a. A node can be added to a cluster resource group even if it has no device entries. Device entries must be added using the Add Cluster Resource Group Device Entry API before the cluster resource group can be started.
    b. If a node is being added to a cluster resource group, the node must be in the same device domain as the other nodes in the recovery domain.
    c. The configuration objects for the device resources in the cluster resource group must exist on the node being added and the resource names in the configuration objects must be the same as the resource names used by the configuration objects on the existing nodes in the recovery domain.

The node being added must be able to access the hardware resources represented by the configuration objects in the cluster resource group.

d. If a data base name is specified in the configuration objects in the cluster resource group, it must be the same on the node being added.

e. If adding a new primary node and an auxiliary storage pool group already exists for the cluster resource group, all members of the auxiliary storage pool group must be configured in the cluster resource group before ownership can be changed.

f. If a server takeover IP address is specified in the cluster resource group and the cluster resource group is active, the server takeover IP address must exist.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

Cluster Resource Group Authority
*CHANGE

Cluster Resource Group Library Authority
*EXECUTE

Cluster Resource Group Lock
*EXCL

Exit Program Authority » (applies to user profile calling the API and user profile to run the exit program) «
*EXECUTE

Exit Program Library Authority » (applies to user profile calling the API and user profile to run the exit program) «
*EXECUTE

User Profile Authority » (applies to user profile to run the exit program) «
*USE

Failover Message Queue Authority
*OBJOPR, *ADD

Failover Message Queue Library Authority
*EXECUTE

Request Information User Queue Authority
*OBJOPR, *ADD

Request Information User Queue Library Authority
*EXECUTE

Request Information User Queue Lock
*EXCLRD

Configuration Object Authority
*USE and *OBJMGT

Distribute Information User Queue Authority
*OBJOPR, *ADD

Distribute Information User Queue Library Authority
*EXECUTE

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster in which the cluster resource group exists.

Cluster resource group name
INPUT; CHAR(10)

The name of the cluster resource group that will have the new node added to its recovery domain.

Node id
INPUT; CHAR(8)

A unique string of characters that identifies the node being added to the recovery domain of the cluster resource group specified. The node specified must be in the cluster and must be unique in the recovery domain of the cluster resource group specified.

Node role
INPUT; BINARY(4)

The role the node will have in the recovery domain. For primary-backup model cluster resource groups a node can have one of three roles: primary, backup, or replicate. Only one node can be designated as the primary. Backup nodes are assigned a backup order. One indicates the first backup, two the second backup, and so on. Replicates are not ordered and cannot become a primary or backup node unless the "Change Cluster Resource Group (QcstChangeClusterResourceGroup) API" on page 86 is used to change its role from replicate to either a backup or primary.

For peer model cluster resource groups a node can have one of two roles: peer or replicate. Any number of nodes can be designated as a peer or replicate. Peer nodes are not ordered and can be an active access point for the cluster resources. Replicates are not ordered and cannot become an active access point for the cluster resource unless the "Change Cluster Resource Group (QcstChangeClusterResourceGroup) API" on page 86 is used to change its role from replicate to peer.

The following summarizes the valid values for this field:

0 Primary node. The new node is to be the primary node. The cluster resource group must have a status of Inactive (20).

>=1 Backup node. The number indicates the backup order. If there is already a node with the same backup order, the new node is inserted in the position requested. At the completion of the request the nodes with backup roles will be sequentially renumbered from the first backup to the last. The first backup will always be 1.

-1 Replicate node. All replicates have this value. Replicate nodes are not ordered.

-2 Last Backup. The new node id will be added as the last backup.

Peer node. All peers have this value. Peer nodes are not ordered.

-4

Results information
INPUT; CHAR(30)

This parameter identifies a qualified user queue field and is followed by a reserved field.

Qualified user queue: Completion information is returned to this user queue, which exists on the node from which the API was called, after the function has completed. See the "Usage Notes" on page 85 section of this API for a description of the data that is placed on this queue. This is a 20-character field. The first 10 characters contain the user queue name, and the second 10

characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, *CURLIB are not valid library names. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of the 30-character results information are reserved. Each character in this field must be set to hexadecimal zero.

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See "Cluster APIs Use of User Queues" on page 221 and "Using Results Information" on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Error Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF18BA D	Error occurred with subsystem.
CPF2113 E	Cannot allocate library &1.
CPF2204 D	User profile &2 not found.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPF9801 D	Object &2 in library &3 not found.
CPF9802 D	Not authorized to object &2 in &3.
CPF9803 D	Cannot allocate object &2 in library &3.
CPF9804 D	Object &2 in library &3 damaged.
CPF9810 D	Library &1 not found.
CPFBB09 D	Node Id &1 does not exist in cluster &2.
CPFBB0A D	Cluster node &1 in cluster&2 not active.
CPFBB0B D	Request using takeover IP address &1 failed.
CPFBB0F D	Cluster resource group &1 does not exist in cluster &2.
CPFBB17 D	&1 API cannot be processed in cluster &2.
CPFBB18 D	Request &1 not allowed for cluster resource group &2.
CPFBB2C D	Attributes of exit program &1 in library &2 are not valid.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB2E D	Job submission failed for cluster resource group &1 in cluster &2.
CPFBB32 D	Attributes of user queue &1 in library &2 are not valid.
CPFBB35 D	The user profile name &1 is not valid for this request.
CPFBB38 D	Library name &1 not allowed on this request.
CPFBB39 D	Current user does not have IOSYSCFG special authority.
CPFBB46 D	Cluster Resource Service internal error.
CPFBB52 D	Cluster node &1 could not be added to cluster resource group &2.
CPFBB5B D	Resource name &1 incorrect for configuration object &2 on node &3.
CPFBB65 D	Cluster node &1 in different device domain.
CPFBB66 D	Request failed for device cluster resource group &3.
CPFBB67 D	Node &1 cannot take ownership of configuration object &2.
CPFBB6C D	Hardware configuration is not complete for configuration objects in cluster resource group &1.
CPFBB70 D	API request &1 not compatible with current cluster version.
CPFBB7B D	Device type not correct for configuration object &1 on node &2.
CPFBB80 D	Request failed for device cluster resource group &3.
CPFBB81 D	New primary node &1 not active.

Message ID	Error Message Text
CPFBB90 D	Request failed for device cluster resource group &3.
CPFBB92 D	Hardware resource &1 not owned by node &3 or node &4.
CPFBB98 D	Hardware resource &1 not switchable.
CPFBB99 D	Request failed for device cluster resource group &3.
CPFBB9B D	Auxiliary storage pool group member &1 not specified.
CPFBB9E D	Data base name &1 not correct for configuration object &2 on node &3.
CPIBB10 D	Cluster resource group exit program &1 in library &2 on node &3 failed.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the "Usage Notes" on page 85 above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C1E E	Required parameter &1 omitted.
CPF3C39 E	Value for reserved field not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB09 E	Node &1 does not exist in cluster &2.
CPFBB0A E	Cluster node &1 in cluster&2 not active.
CPFBB0F E	Cluster resource group &1 does not exist in cluster &2.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB29 E	Node role value &1 not valid.
CPFBB2C E	Attributes of exit program &1 in library &2 are not valid.
CPFBB32 E	Attributes of user queue &1 in library &2 are not valid.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB44 E	&1 API cannot be called from a cluster resource group exit program.
CPFBB52 E	Cluster node &1 not added to cluster resource group &2.

API introduced: V4R4

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Change Cluster Resource Group (QcstChangeClusterResourceGroup) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Cluster resource group name	Input	Char(10)
4	Cluster resource group description information	Input	Char(*)
5	Format name	Input	Char(8)

6	Text description	Input	Char(50)
7	Results information	Input	Char(30)
8	Error code	I/O	Char(*)

Service Program: QCSTCRG1
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Change Cluster Resource Group (QcstChangeClusterResourceGroup) API changes some of the attributes of a cluster resource group.

The following fields may be changed without causing the cluster resource group exit program to be called:

- Text description
- Exit program data
- User profile
- Takeover IP address
- Job name
- Allow application restart
- Number of restarts
- The cluster resource group exit program
- Failover message queue
- Failover wait time
- Failover default action
- Cluster resource group exit program format name
- Application id

To add a node to the recovery domain use, the "Add Node To Recovery Domain (QcstAddNodeToRcvyDomain) API" on page 81.

To remove a node from the recovery domain, use the "Remove Node From Recovery Domain (QcstRemoveNodeFromRcvyDomain) API" on page 160.

To add a device to a resilient device cluster resource group, use the "Add Cluster Resource Group Device Entry (QcstAddClusterResourceGroupDev) API" on page 75.

To remove a device from a resilient device cluster resource group, use the "Remove Cluster Resource Group Device Entry (QcstRmvClusterResourceGroupDev) API" on page 156.

To change a device entry in a resilient device cluster resource group, use the "Change Cluster Resource Group Device Entry (QcstChgClusterResourceGroupDev) API" on page 102.

This API will do the following for all cluster resource group types:

- Call the cluster resource group exit program with an action code of Change (13) on all active nodes in the recovery domain when either the preferred or current role is changed,or when a site name and/or data port IP addresses are changed, if an exit program is specified for the cluster resource group. The cluster resource group status is set to Change Pending (520). If the exit program completes successfully, the cluster resource group status is reset to its value at the time the API was called. If the exit program fails and the cluster resource group cannot be restored to its original condition, the cluster resource group status is set to Indoubt (30).
- Change the cluster resource group without calling the exit program if neither node role, site name nor data port IP addresses are changed.

- Change the name to be used for batch jobs submitted by cluster resource group. If the cluster resource group status is Active (10), batch jobs already submitted will not be changed. Any jobs submitted after the change will use the new name. This is true for other attributes associated with a submitted exit program such as the user profile, the restart count and so on. Changes to the cluster resource group will not affect an exit program that was previously submitted and is either on a job queue or is running.
- » For a primary-backup model cluster resource group, « if the current node role in the recovery domain is changed and the cluster resource group is active and it has more than one backup node and some backup nodes are not active, the recovery domain may be reordered so that all active backup nodes are ordered before inactive backup nodes.

This API will do the following for resilient application cluster resource groups:

- If the Cluster Resource Services configures the takeover IP address, it will remove the current address and add the new address when the takeover IP address is changed. If either the add or remove address function fails, the API will fail.
- If the cluster resource group is active and the role of a node is being changed from replica to backup, verify the takeover IP address exists and is not active on the node being changed. If the takeover IP address does not exist or is active on the node being changed, the API will fail.

This API will do the following for resilient device cluster resource groups:

- If the role of the current primary node is being changed, ownership of the devices specified in the cluster resource group is switched from the current primary to the new primary if the current primary has none of the devices varied on. If any devices are varied on, an error message is returned. In addition, the new primary node must be active. All members of an auxiliary storage pool group must be configured in the cluster resource group before ownership can be changed. Devices are not varied on after the ownership is switched.
- If the site name and/or data port IP addresses for a recovery domain node is changed but the node role is not changed, you can specify only the node that is changed, no need to specify all nodes in the recovery domain.
- If a recovery domain node role is changed (a value other than -2 is specified), then the rest of the nodes in the recovery domain must also be specified and they must have value other than -2.
- In cross-site mirroring, a switchover on a mirror site can be done by changing the mirror site node roles, by assigning a different active node to be the highest node role ranking at a mirror site.

This API requires for all cluster resource group types:

1. Cluster Resource Services must be active on the node processing the request.
2. A cluster resource group must have a status of Inactive (20) or Indoubt (30) to designate a new primary node.
3. The new exit program, if one specified, must exist on all nodes in the recovery domain when the cluster resource group exit program is changed.
4. At least one active node in the recovery domain.
5. If defined, the failover message queue must exist on all nodes in the recovery domain when the cluster resource group is changed.

» This API requires for all primary-backup model cluster resource group types: «

1. Changing the node role to primary or changing the takeover IP address can only be done when the cluster resource group is Inactive (20) or Indoubt (30). If the cluster resource group is Active, the Initiate Switchover API can be used to assign the primary role to the first backup node. For information about the Initiate Switchover API, see “Initiate Switchover (QcstInitiateSwitchOver) API” on page 135.

» This API requires for all peer model cluster resource group types:

1. The recovery domain role can be changed from peer to replicate or replicate to peer by specifying one or more recovery domain nodes. The full recovery domain does not need to be specified.
2. If the cluster resource group is Active (10) and:
   - The role is changed from peer to replicate, the node becomes an inactive access point.
   - The role is changed from replicate to peer, the node becomes an active access point.
3. There must be at least one node designated as peer if the cluster resource group is Active (10). «

This API operates in an asynchronous mode. See "Behavior of Cluster Resource Services APIs" on page 221 for more information.

Restriction: This API cannot be called from a cluster resource group exit program.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

Cluster Resource Group Authority
*CHANGE

Cluster Resource Group Library Authority
*EXECUTE

Cluster Resource Group Group Lock
*EXCL

Exit Program Authority » **(applies to user profile calling the API and user profile to run the exit program)** «
*EXECUTE

Exit Program Library Authority » **(applies to user profile calling the API and user profile to run the exit program)** «
*EXECUTE

User Profile Authority » **(applies to user profile to run the exit program)** «
*USE

Request Information User Queue Authority
*OBJOPR, *ADD

Request Information User Queue Library Authority
*EXECUTE

Request Information User Queue Lock
*EXCLRD

Configuration Object Authority
*USE and *OBJMGT

Failover Message Queue Authority
*OBJOPR, *ADD

Failover Message Queue Library Authority
*EXECUTE

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster to which the cluster resource group belongs.

Cluster resource group name
INPUT; CHAR(10)

The name of the cluster resource group which is to be changed.

Cluster resource group description information
INPUT; CHAR(*)

Detailed information about the cluster resource group. For more information, see "Data Resiliency (RGDC0100 Format)" on page 91, "Data Resiliency (RGDC0110 Format)" on page 91, "Application Resiliency (RGDC0200 Format)" on page 92, "Device Resiliency (RGDC0300 Format)" on page 93 » and Peer Resiliency (RGDC0400 Format) (page "Peer Resiliency (RGDC0400 Format)" on page 94)« .

Format name
INPUT; CHAR(8)

The content and format of the cluster resource group information. The possible format names are:

"Data Resiliency (RGDC0100 Format)" on page 91	This format describes the data cluster resource group.
"Data Resiliency (RGDC0110 Format)" on page 91	This format also describes the data cluster resource group, with some additional fields added.
"Application Resiliency (RGDC0200 Format)" on page 92	This format describes the application cluster resource group.
"Device Resiliency (RGDC0300 Format)" on page 93	This format describes the device cluster resource group.
» RGDC0400 (page "Peer Resiliency (RGDC0400 Format)" on page 94)	This format describes the peer cluster resource group. «

Text description
INPUT; CHAR(50)

This text briefly describes the cluster resource group. This must be left justified. The following special value can be used:

**SAME*	The current text description is not changed. This must be left justified.

Results information
INPUT; CHAR(30)

This parameter identifies a qualified user queue field and is followed by a reserved field.

Qualified user queue: Completion information is returned to this user queue, which exists on the node from which the API was called, after the function has completed. See the “Usage Notes” on page 99 section of this API for a description of the data that is placed on this queue. This is a 20-character field. The first 10 characters contain the user queue name, and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, *CURLIB are not valid library names. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of the 30-character results information are reserved. Each character in this field must be set to hexadecimal zero.

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Data Resiliency (RGDC0100 Format)

Offset		Type	Field
Dec	**Hex**		
0	0	CHAR(10)	Cluster resource group exit program name
10	A	CHAR(10)	Cluster resource group exit program library name
20	14	CHAR(8)	Cluster resource group exit program format name
28	1C	CHAR(10)	User profile
38	26	CHAR(10)	Action for recovery domain array
48	30	CHAR(256)	Exit program data
304	130	BINARY(4)	Offset to recovery domain array
308	134	BINARY(4)	Number of nodes in recovery domain
*	*	Array (*) of CHAR(*)	Recovery domain array
These fields repeat, in the order listed, for each node in the recovery domain.		CHAR(8)	Node id
		BINARY(4)	Node role

Data Resiliency (RGDC0110 Format)

Offset		Type	Field
Dec	**Hex**		
0	0	CHAR(10)	Cluster resource group exit program name
10	A	CHAR(10)	Cluster resource group exit program library name
20	14	CHAR(8)	Cluster resource group exit program format name
28	1C	CHAR(10)	User profile
38	26	CHAR(10)	Action for recovery domain array
48	30	CHAR(256)	Exit program data
304	130	BINARY(4)	Offset to recovery domain array
308	134	BINARY(4)	Number of nodes in recovery domain
312	138	BINARY(4)	Failover wait time
316	13C	BINARY(4)	Failover default action
320	140	CHAR(10)	Failover message queue name

Offset		Type	Field
Dec	Hex		
330	14A	CHAR(10)	Failover message queue library name
340	154	BINARY(4)	Offset to additional fields
344	158	BINARY(4)	Length of additional fields
*	*	Array (*) of CHAR(*)	Recovery domain array
These fields repeat, in the order listed, for each node in the recovery domain.		CHAR(8)	Node id
		BINARY(4)	Node role

Application Resiliency (RGDC0200 Format)

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Cluster resource group exit program name
10	A	CHAR(10)	Cluster resource group exit program library name
20	14	CHAR(8)	Cluster resource group exit program format name
28	1C	CHAR(10)	User profile
38	26	CHAR(10)	Action for recovery domain array
48	30	CHAR(256)	Exit program data
304	130	BINARY(4)	Offset to recovery domain array
308	134	BINARY(4)	Number of nodes in recovery domain
312	138	CHAR(16)	Application takeover IP address
328	148	CHAR(10)	Job name
338	152	CHAR(1)	Additional fields used
339	153	CHAR(1)	Reserved
340	154	BINARY(4)	Allow application restart
344	158	BINARY(4)	Number of restarts
348	15C	BINARY(4)	Offset to additional fields
352	160	BINARY(4)	Length of additional fields
*	*	Array (*) of CHAR(*)	Recovery domain array
These fields repeat, in the order listed, for each node in the recovery domain.		CHAR(8)	Node id
		BINARY(4)	Node role
*	*	CHAR(28)	Additional fields
These fields are part of the additional fields structure.		BINARY(4)	Failover wait time
		BINARY(4)	Failover default action
		CHAR(10)	Failover message queue name
		CHAR(10)	Failover message queue library name

Device Resiliency (RGDC0300 Format)

Offset		Type	Field
Dec	**Hex**		
0	0	CHAR(10)	Cluster resource group exit program name
10	A	CHAR(10)	Cluster resource group exit program library name
20	14	CHAR(8)	Cluster resource group exit program format name
28	1C	CHAR(10)	User profile
38	26	CHAR(10)	Action for recovery domain array
48	30	CHAR(256)	Exit program data
304	130	BINARY(4)	Offset to recovery domain array
308	134	BINARY(4)	Number of nodes in recovery domain
312	138	BINARY(4)	Length of recovery domain array entry
316	13C	BINARY(4)	Offset to additional fields
320	140	BINARY(4)	Length of additional fields
*	*	Array (*) of CHAR(*)	Recovery domain array if Length of recovery domain array entry field is set to non-zero
These fields repeat, in the order listed, for each node in the recovery domain.		CHAR(8)	Node id
		BINARY(4)	Node role
*	*	Array (*) of CHAR(*)	Recovery domain array if Length of recovery domain array entry field is set to zero
These fields repeat, in the order listed, for each node in the recovery domain.		BINARY(4)	Length of entry in the recovery domain
		CHAR(8)	Node id
		BINARY(4)	Node role
		CHAR(8)	Site name
		BINARY(4)	Offset to data port IP address array
		BINARY(4)	Number of data port IP addresses
		BINARY(4)	Data Port IP address action
		Array(*) of CHAR(16)	Data Port IP address
*	*	CHAR(28)	Additional fields
These fields are part of the additional fields structure.		BINARY(4)	Failover wait time
		BINARY(4)	Failover default action
		CHAR(10)	Failover message queue name
		CHAR(10)	Failover message queue library name

Peer Resiliency (RGDC0400 Format)

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of fixed fields
4	4	BINARY(4)	Offset to recovery domain array
8	8	BINARY(4)	Number of nodes in recovery domain
12	C	BINARY(4)	Length of recovery domain array entry
16	10	CHAR(10)	Action for recovery domain array
26	1A	CHAR(10)	Cluster resource group exit program name
36	24	CHAR(10)	Cluster resource group exit program library name
46	2E	CHAR(8)	Cluster resource group exit program format name
54	36	CHAR(10)	User profile
64	40	CHAR(256)	Exit program data
320	140	CHAR(20)	Application id
*	*	Array (*) of CHAR(*)	Recovery domain array
These fields repeat, in the order listed, for each node in the recovery domain.		CHAR(8)	Node id
		BINARY(4)	Node role «

Field Descriptions

Action for recovery domain array. Indicates which node role in the recovery domain is being changed. The special values used are:

**SAME*	Neither node role is changed.
**CHGPREFER*	The **PREFFERED** node role in the recovery domain will change.
**CHGCURREN*	The **CURRENT** node role in the recovery domain can be changed. For a device cluster resource group, site name and data port IP addresses also can be changed.

Additional fields. A structure containing optional additional fields.

Additional fields used. A flag to signify whether the additional fields in format RGDC0200 are being used. If the cluster version is less than 3, this field must be set to hexadecimal zero. Possible values are:

0x00	The additional fields are not being used.
0x01	The additional fields are being used.

Allow application restart. Attempt to restart an application if the cluster resource group exit program fails. Possible values are:

0	Do not attempt to restart the application. The cluster resource group exit program is called with an action code of Failover (9).
1	Attempt to restart the application on the same node. The cluster resource group exit program will be called with an action code of Restart (3). If the application cannot be restarted in the specified maximum number of attempts, the cluster resource group exit program will be called with an action code of Failover (9).
-1	Allow application restart is not changed.

» Application id. This is a string to identify the application that supplied the cluster resource group. The recommended format is 'vendor-id.name' where vendor-id is an identifier for the vendor creating the cluster resource group and name is the application name. For example, QIBM.ExamplePeer, indicates it is supplied by IBM for the ExamplePeer application. It is not recommended to use QIBM for vendor id name unless the cluster resource group is supplied by IBM. «

**SAME* The application id is not changed. This must be left justified.

Application takeover IP address. This is the floating IP address that is to be associated with an application. This field must be represented in dotted decimal format and be a null-terminated string. The following special value can be used:

**SAME* The takeover IP address is not changed. This must be left justified.

If the value is not *SAME and the Cluster Resource Services configured the takeover IP address, this API will remove the current takeover IP address and add this takeover IP address to the node. If either the add or remove address function fails, the API will fail. The cluster resource group must be Inactive (20) to change this field.

Cluster resource group exit program format name. The contents and format of the cluster resource group exit program information. This field must be set to hexadecimal zeroes if no exit program is specified. If the exit program name is *SAME and was previously *NONE, this field is ignored. The format name supported is:

EXTP0100 Exit program information. » This value is allowed for primary-backup model cluster resource group and peer model cluster resource group. «

EXTP0200 Exit program information, with additional information in the recovery domain array which contains site name and data port IP addresses on each node. » This value is allowed for primary-backup model cluster resource group. «

**SAME* Exit program format name is not changed.

Cluster resource group exit program library name. The name of the library where the exit program exists. The special value *CURLIB or *LIBL may not be used for the library name. QTEMP is not a valid library name.

If the cluster resource group exit program name is *NONE or *SAME, the exit program library name is ignored.

Cluster resource group exit program name. The name of exit program that is used to handle action codes that are passed to it. The exit program cannot be in an independent auxiliary storage pool.

If the exit program is changed for an active application cluster resource group, the job currently running which was submitted to handle the Start (2) action code continues running the prior exit program.

The following special value can be used:

**SAME* The current exit program is not changed. This must be left justified.

**NONE* The cluster resource group does not have an exit program.This is valid only for a device. This must be left justified.

Data port IP address. The IP address associated with the recovery domain node. When adding a data port IP address, it must already exist on the specified node if the CRG is active. User is responsible for starting/ending data port IP address. This field must be represented in dotted decimal format and be a null-terminated string.

Data port IP address action. Indicates whether to add or remove the data port IP address associated with the recovery domain node. The possible values are:

-1	The data port IP addresses are not changed.
0	Remove the data port IP addresses from the recovery domain node.
1	Add the data port IP addresses to the recovery domain node.

Exit program data. 256 bytes of data that is passed to the cluster resource group exit program when it is called. This parameter may contain any scalar data except pointers. For example, it can be used to provide state information. This data will be stored with the specified cluster resource group and copied to all nodes in the recovery domain. Pointers in this area will not resolve correctly on all nodes and should not be placed in the data. See "Cluster Resource Group Exit Program" on page 185 for information about the cluster resource group exit program. The data specified will replace the existing exit program data stored with the cluster resource group, if the API completes successfully. This field must be set to hexadecimal zeroes if no exit program is specified. If the exit program name is *SAME and was previously *NONE, this field is ignored.

The following special value can be used:

**SAME*	The exit program data is not changed. This must be left justified.

Failover default action. Indicates what clustering should do pertaining to the failover request when a response to the failover message queue was not be received in the failover wait time limit. If the failover message queue is *NONE, this field must be set to 0. If the failover message queue is *SAME and was previously *NONE, this field must be set to -1 or 0.

-1	Failover default action is not changed. This is the default value.
0	Proceed with failover.
1	Do NOT attempt failover.

Failover message queue library name. The name of the library that contains the user queue to receive failover messages. The library name cannot be *CURLIB, QTEMP, or *LIBL.

If the failover message queue name is *NONE or *SAME, the failover message queue library name is ignored. Version 2 or lower cluster resource groups will default to *NONE for the message queue library name when the cluster version is changed to 3.

Failover message queue name. The name of the message queue to receive messages dealing with failover. The queue cannot be in an independent auxiliary storage pool.

The following special values can be used:

**SAME*	The current failover message queue is not changed. This is the default value. This must be left justified.
**NONE*	No messages will be sent when a failover occurs through this cluster resource group. This must be left justified.

Failover wait time. Number of minutes to wait for a reply to the failover message that was enqueued on the failover message queue. If the failover message queue is *NONE, this field must be set to 0. If the failover message queue is *SAME and was previously *NONE, this field must be set to -2 or 0. If a failover message queue is specified, this field cannot be set to 0. Valid values are:

-2	Failover wait time is not changed. This is the default value.
-1	Wait forever until a response is given to the failover inquiry message.

0	Failover proceeds without user intervention. Acts the same as V5R1M0 and prior.
>=1	Number of minutes to wait for a response to the failover inquiry message. If no response is received in the specified number of minutes, the failover default action field will be looked at to decide how to proceed.

Job name. The name given the batch job that is submitted by the cluster resource group. This job will call the cluster resource group exit program with the action code generated by the API being used. If this field is blank, the job name will be the value of the job description found in the user profile. Valid special values are:

**SAME*	The job name is not changed. This must be left justified.
**JOBD*	The job name in the job description for the specified user profile will be used. This must be left justified.

Length of additional fields. The length in bytes of additional fields. In format RGDC0200, this field is ignored if the additional fields used flag is not set to 1. If the additional fields used flag is 1 in format RGDC0200, the value of this field must be less than or equal to 28. In format RGDC0300, the value of this field must be less than or equal to 28. In format RGDC0110 this field must be set to hexadecimal zero. It will be used in a future release if more fields are needed.

Length of entry in the recovery domain. The length of an entry in the recovery domain array. This field is used if each entry may have a different length.

» Length of fixed fields. The length of the fixed fields in the format description. For RGDC0400 this value must be 340. «

Length of recovery domain array entry. The length of an entry in the recovery domain array. For formats other than RGDC0300, this field must be set to 12. For format RGDC0300, this field can be set to either 12 or zero. If zero, then the length of entry in the recovery domain field is used.

Node id. A unique string of characters that identifies a node that is participating in the recovery domain of the specified cluster resource group. The node specified must be active in the cluster, and it must be unique in the recovery domain of the specified cluster resource group.

Node role. The role the node has in the recovery domain. A role must be defined for each node in the recovery domain. » For primary-backup model cluster resource groups « a node can have one of three roles: primary, backup, or replicate. Only one node can be designated as the primary. Backup nodes are assigned a backup order. One indicates the first backup, two the second backup, and so on. Replicates are not ordered and cannot become a primary or backup node unless its role is changed from replicate to either a backup or primary.
» For peer model cluster resource groups a node can have one of two roles: peer or replicate. Any number of nodes can be designated as the peer or replicate. Peer nodes are not ordered and can be an active access point for the cluster resources. Replicates are not ordered and cannot become an active access point for the cluster resource unless its role is changed from replicate to peer.
« The following summarizes the valid values for this field:

0	Primary node. Only one node can have this value.
>=1	Backup node. The backup order is designated by increasing value. The values need not be consecutive. No two backup nodes can have the same value. At the completion of the API, Cluster Resource Services will sequence the backups using consecutive numbers starting with 1.
-1	Replicate node. All replicates have this value.
-2	The node role is not changed. »
-4	Peer node. All peers have this value. «

Number of data port IP addresses. The number of data port IP addresses to be added to or removed from the recovery domain node. If the current cluster version is 3 and the length of recovery domain array specified includes this field, it must be set to zero. This field must also be set to zero if no change is made to the data port IP addresses.

Number of nodes in the recovery domain. The number of nodes in the recovery domain array. This field is ignored if the Action for recovery domain array field contains *SAME. » For primary-backup model cluster resource groups « this should equal the number of backup nodes plus the number of replicate nodes plus one (for the primary node). This must be greater than or equal to one and equal the number of nodes in the recovery domain. » For peer model cluster resource groups all recovery domain nodes do not need to be specified to change the role. Only specify those nodes that are changing. If roles are changing this must be equal to the number of nodes being changed and cannot exceed the number of nodes currently defined in the recovery domain. «

Number of restarts. Number of times a cluster resource group exit program can be called on a same node before failure occurs. Maximum number of restarts is 3. -1 means the maximum number of restarts does not change. If the cluster resource group is currently active, any change does not take affect until failover occurs or the cluster resource group exit program job ends.

Offset to additional fields. The byte offset from the beginning of this parameter to additional fields. In format RGDC0200, this field will be ignored unless the additional fields used field is set to 1. In format RGDC0110, this must be set to hexadecimal zero.

Offset to data port IP address array. The byte offset from the beginning of this parameter to the first data port IP address. If the current cluster version is 3 and the length of recovery domain array specified includes this field, it must be set to zero. This field is ignored if the number of data port IP addresses field is set to zero.

Offset to recovery domain array. The byte offset from the beginning of this table to the array of node information. This field is ignored if the Action for recovery domain array field contains *SAME.

Recovery domain array. » Array of recovery domain information. A role must be defined for each node specified. Nodes in the recovery domain must be unique. See node role for more information on the node roles associated with the specific cluster resource group type. For primary-backup model cluster resource groups this array identifies the nodes that compose the recovery domain. The complete recovery domain must be specified. For peer model cluster resource groups, this array identifies the recovery domain nodes that are changing. Only those nodes that are being changed need to be specified. «

An example » of a primary-backup model cluster resource group «: A cluster resource group has four nodes: NodeA, NodeB, NodeC and NodeD. NodeA is the primary. There are two backup nodes: NodeB and NodeD. NodeD is the first backup and NodeB is the second backup. There is one replicate: NodeC.

```
Node   Role
-----  ----
NodeA   0   <-- primary
NodeD   1   <-- backup #1
NodeB   2   <-- backup #2
NodeC   -1  <-- replicate
```

The nodes do not have to be arranged in any particular order in the array. They could be in the array as listed below and have the same result.

```
Node   Role
-----  ----
NodeB   2   <-- backup #2
NodeA   0   <-- primary
NodeC   -1  <-- replicate
NodeD   1   <-- backup #1
```

Reserved. Must contain hexadecimal zeroes.

Site name. The name of the site associated with the recovery domain node. The site name can be changed from *NONE to a name, or from a name to *NONE. Changing a name to a different name is not allowed. If the current cluster version is 3 and the length of recovery domain array specified includes this field, it must be set to hexadecimal zeroes. Valid special values for this field are:

**NONE*	The node in the recovery domain is not associated with a site name. This must be left justified.
**SAME*	The site name it not changed. This must be left justified.

User profile. The name of the user profile under which the exit program should process. The user profile must exist on all nodes in the recovery domain. This field must be set to hexadecimal zeroes if no exit program is specified. If the exit program name is *SAME and was previously *NONE, this field is ignored.

The following user profiles are not valid:

- QDBSHR
- QDOC
- QDTFOWN
- QRJE
- QLPAUTO
- QLPINSTALL
- QSECOFR
- QSPL
- QSYS
- QTSTRQS

The following special value can be used:

**SAME*	The current user profile is not changed. This must be left justified.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See “Cluster APIs Use of User Queues” on page 221 and “Using Results Information” on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF18BA D	Error occurred with subsystem.
CPF2113 E	Cannot allocate library &1.
CPF2204 D	User profile &1 not found.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPF9801 D	Object &2 in library &3 not found.
CPF9802 D	Not authorized to object &2 in &3.
CPF9803 D	Cannot allocate object &2 in library &3.
CPF9804 D	Object &2 in library &3 damaged.

Message ID	Message Text
CPF9810 D	Library &1 not found.
CPFBB0B D	Request using takeover IP address &1 failed.
CPFBB0F D	Cluster resource group &1 does not exist in cluster &2.
CPFBB17 D	&1 API cannot be processed in cluster &2.
CPFBB18 D	Request &1 is not allowed for cluster resource group &2.
CPFBB1A D	At least one node in the recovery domain of cluster resource group &1 must be active.
CPFBB2C D	Attributes of exit program &1 in library &2 are not valid.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB2E D	Job submission failed for cluster resource group &1 in cluster &2.
CPFBB32 D	Attributes of user queue &1 in library &2 are not valid.
CPFBB35 D	The user profile name &1 is not valid for this request.
CPFBB38 D	Library name &1 is not allowed for this request.
CPFBB39 D	The current user does not have IOSYSCFG special authority.
CPFBB46 D	Cluster Resource Services internal error.
CPFBB51 D	IP address &1 already in use by the cluster &3.
CPFBB5E E	User profile to run exit program not specified.
CPFBB66 D	Request failed for device cluster resource group &3.
CPFBB67 D	Node &1 cannot take ownership of configuration object &2.
CPFBB69 D	Primary node &1 not current owner of hardware resource &2.
CPFBB6C D	Hardware configuration is not complete for configuration objects in cluster resource group &1.
CPFBB70 D	API request &1 not compatible with current cluster version.
CPFBB80 D	Request failed for device cluster resource group &3.
CPFBB81 D	New primary node &1 not active.
CPFBB90 D	Request failed for device cluster resource group &3.
CPFBB92 D	Hardware resource &1 not owned by node &3 or node &4.
CPFBB98 D	Hardware resource &1 not switchable.
CPFBB99 D	Request failed for device cluster resource group &3.
CPFBB9B D	Auxiliary storage pool group member &1 not specified.
CPFBBA2 D	Value &1 specified for failover wait time is not valid.
CPFBBA3 D	Value &1 specified for failover default action is not valid.
CPFBBA4 D	Field value within additional fields structure is not valid.
CPFBBA7 D	Site name and data port IP address not match.
CPFBBA8 D	Site name &1 specified for node &2 not allowed.
CPFBBA9 D	Data port IP address &1 specified for node &2 not allowed.
CPFBBB0 D	Exit program format name not specified.
CPIBB10 D	Cluster resource group exit program &1 in library &2 on node &3 failed.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the “Usage Notes” on page 99 above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF2204 E	User profile &1 not found.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C29 E	Object name &1 is not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.

Message ID	Error Message Text
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB09 E	Node Id &1 is not a member of Cluster &2.
CPFBB0A E	Cluster node &1 in cluster &2 not active.
CPFBB0F E	Cluster resource group &1 does not exist in cluster &2.
CPFBB25 E	Value &1 specified for recovery domain array action is not valid.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB27 E	A primary node was not specified for the recovery domain.
CPFBB28 E	Cluster node &1 and cluster node &2 have the same node role value &3.
CPFBB29 E	Node role value &1 not valid.
CPFBB2C E	Attributes of exit program &1 in library &2 are not valid.
CPFBB30 E	Takeover IP address &1 not part of the TCP/IP subnet.
CPFBB31 E	Value &1 specified for number of restarts not valid.
CPFBB32 E	Attributes of user queue &1 in library &2 not valid.
CPFBB33 E	Cluster node &1 already exists in recovery domain for cluster resource group &4.
CPFBB35 E	The user profile name &1 is not valid for this request.
CPFBB36 E	The number of cluster nodes specified for the recovery domain is not valid.
CPFBB37 E	The offset to the recovery domain array is not valid.
CPFBB38 E	Library name &1 is not allowed for this request.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB40 E	The value &1 specified for the allow application restarts parameter is not valid.
CPFBB43 E	Invalid format name &1 for cluster resource group type &2.
CPFBB44 E	&1 API cannot be called from a cluster resource group exit program.
CPFBB51 E	IP address &1 already in use by the cluster &3.
CPFBB5E E	User profile to run exit program not specified.
CPFBB5F E	Field value within structure is not valid.
CPFBB62 E	Exit program name *NONE not valid.
CPFBB70 E	API request &1 not compatible with current cluster version.
CPFBBA2 E	Value &1 specified for failover wait time is not valid.
CPFBBA3 E	Value &1 specified for failover default action is not valid.
CPFBBA7 E	Site name and data port IP address not match.
CPFBBA8 E	Site name &1 specified for node &2 not allowed.
CPFBBA9 E	Data port IP address &1 specified for node &2 not allowed.
CPFBBAC E	The offset to the data port IP address array for node &1 is not valid.
CPFBBAD E	The number of data port IP addresses specified for node &1 is not valid.
CPFBBAE E	The data port IP address action for node &1 is not valid.
CPFBBAF E	Recovery domain node role is not valid.
CPFBBB0 E	Exit program format name not specified.
TCP1901 E	Internet address &1 not valid.

API introduced: V4R4

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Change Cluster Resource Group Device Entry (QcstChgClusterResourceGroupDev) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Cluster resource group name	Input	Char(10)
4	Configuration object entry information	Input	Char(*)
5	Format name	Input	Char(8)
6	Results information	Input	Char(30)
7	Error code	I/O	Char(*)

Service Program: QCSTCRG1
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Change Cluster Resource Group Device Entry (QcstChgClusterResourceGroupDev) API changes information about one or more configuration objects in a device cluster resource group. The entry being changed is found by searching the current entries for a matching configuration object name and configuration object type.

If an exit program is specified for the cluster resource group, the cluster resource group exit program is called with an action code of Change Device Entry (19) on all active nodes in the recovery domain. The cluster resource group status is set to Change Device Entry Pending (610). If the exit program completes successfully, the cluster resource group status is reset to its value at the time the API was called. If the exit program fails and the cluster resource group cannot be restored to its original condition, the cluster resource group status is set to Indoubt (30).

This API requires:

1. Cluster Resource Services must be active on the node processing the request.
2. The number of configuration object entries in the configuration object array cannot exceed 256.
3. If an exit program is specified, the exit program must exist on all nodes in the recovery domain.
4. At least one node in the recovery domain must be active.
5. If a server takeover IP address is specified, it must exist on all nodes in the recovery domain if the cluster resource group is active. The server takeover IP address must be unique. It can only be associated with a primary auxiliary storage pool.

This API operates in an asynchronous mode. See “Behavior of Cluster Resource Services APIs” on page 221 for more information.

Restriction: This API cannot be called from a cluster resource group exit program.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

Cluster Resource Group Authority
*CHANGE

Cluster Resource Group Library Authority
*EXECUTE

Cluster Resource Group Lock
*EXCL

Exit Program Authority (applies to user profile calling the API and user profile to run the exit program)
*EXECUTE

Exit Program Library Authority (applies to user profile calling the API and user profile to run the exit program)
*EXECUTE

User Profile Authority (applies to user profile to run the exit program)
*USE

Request Information User Queue Authority
*OBJOPR, *ADD

Request Information User Queue Library Authority
*EXECUTE

Request Information User Queue Lock
*EXCLRD

Configuration Object Authority
*USE and*OBJMGT

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster to which the cluster resource group belongs.

Cluster resource group name
INPUT; CHAR(10)

The name of the cluster resource group which is to be changed.

Configuration object entry information
INPUT; CHAR(*)

Detailed information about the configuration object entries to be changed. For more information, see "Device Resiliency (RGDH0100 Format)" on page 104.

Format name
INPUT; CHAR(8)

The content and format of the device information. The possible format names are:

"Device Resiliency (RGDH0100 Format)" on page 104	This format describes the resilient device.

Results information
INPUT; CHAR(30)

This parameter identifies a qualified user queue field and is followed by a reserved field.

Qualified user queue: Completion information is returned to this user queue, which exists on the node from which the API was called, after the function has completed. See the "Usage Notes" on page 105 section of this API for a description of the data that is placed on this queue. This is a 20-character field. The first 10 characters contain the user queue name, and the second 10

characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, *CURLIB are not valid library names. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of the 30-character results information are reserved. Each character in this field must be set to hexadecimal zero.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Device Resiliency (RGDH0100 Format)

Offset		Type	Field
Dec	**Hex**		
0	0	BINARY(4)	Offset to configuration object array
4	4	BINARY(4)	Number of entries in configuration object array
8	8	BINARY(4)	Length of configuration object array entry
12	C	BINARY(4)	Offset to additional fields
16	10	BINARY(4)	Length of additional fields
*	*	Array (*) of CHAR(36)	Configuration object array
These fields repeat, in the order listed, for each device.		CHAR(10)	Configuration object name
		CHAR(2)	Reserved
		BINARY(4)	Configuration object type
		BINARY(4)	Configuration object online
		CHAR(16)	Server takeover IP address

Field Descriptions

Configuration object array. This array identifies the resilient devices.

Configuration object name. The name of the auxiliary storage pool device description which is in the cluster resource group.

Configuration object online. Vary the configuration object on and start the server takeover IP address or leave the configuration object varied off and the server takeover IP address inactive when a device is switched from one node to another with the “Initiate Switchover (QcstInitiateSwitchOver) API” on page 135 or when it is failed over to a backup node. For secondary auxiliary storage pools, only a value of 2 is valid. If cluster resource services cannot determine if this value is correct for a device entry because the auxiliary storage pool is not yet created, any errors will be detected when the cluster resource group is started. A value of 2 cannot be specified for any other device type. Possible values are:

-1 Configuration object online is not changed.
0 Do not vary the configuration object on and do not start the server takeover IP address.
1 Vary the configuration object on and start the server takeover IP address.
2 Perform the same action for a secondary auxiliary storage pool as is specified for the primary.

Configuration object type. This specifies the type of configuration object specified with configuration object name. Possible values are:

Length of additional fields. The length in bytes of additional fields. This must be set to hexadecimal zero. It will be used in a future release if more fields are needed in the RGDH0100 format.

Length of configuration object array entry. This specifies the length of an entry in the configuration object array.

Number of entries in configuration object array. The number of entries in the configuration object array. This number must be greater than zero and less than or equal to 256.

Offset to additional fields. The byte offset from the beginning of this parameter to additional fields. This must be set to hexadecimal zero. It will be used in a future release if more fields are needed in the RGDH0100 format.

Offset to configuration object array. The byte offset from the beginning of this parameter to the configuration object array field.

Reserved. Must contain hexadecimal zeroes.

Server takeover IP address. This is a takeover IP address for servers associated with the relational database name in the device description for an auxiliary storage pool. This field is optional and can only be specified for a primary auxiliary storage pool. If specified, the address must be represented in dotted decimal format and be a null-terminated string. The specified address must exist on all nodes in the recovery domain if the cluster resource group is active. If not specified, or for a secondary and UDFS auxiliary storage pool, this field must be set to *NONE and be left justified. If the current cluster version is 2 and the length of configuration object array entry specified includes the server takeover IP address, this field must be set to hexadecimal zeroes. Valid special values for this field are:

**SAME*	The server takeover IP address does not change. This must be left justified.
**NONE*	There is no server takeover IP address associated with the relational database name in the device description for an auxiliary storage pool. This must be left justified.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See “Cluster APIs Use of User Queues” on page 221 and “Using Results Information” on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Error Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF18BA D	Error occurred with subsystem.
CPF2113 E	Cannot allocate library &1.
CPF2204 D	User profile &1 not found.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPF9801 D	Object &2 in library &3 not found.
CPF9802 D	Not authorized to object &2 in &3.

Message ID	Error Message Text
CPF9803 D	Cannot allocate object &2 in library &3.
CPF9804 D	Object &2 in library &3 damaged.
CPF9810 D	Library &1 not found.
CPFBB0F D	Cluster resource group &1 does not exist in cluster &2.
CPFBB17 D	&1 API cannot be processed in cluster &2.
CPFBB18 D	Request &1 is not allowed for cluster resource group &2.
CPFBB2C D	Attributes of exit program &1 in library &2 are not valid.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB2E D	Job submission failed for cluster resource group &1 in cluster &2.
CPFBB32 D	Attributes of user queue &1 in library &2 are not valid.
CPFBB35 D	The user profile name &1 is not valid for this request.
CPFBB38 D	Library name &1 is not allowed for this request.
CPFBB39 D	The current user does not have IOSYSCFG special authority.
CPFBB46 D	Cluster Resource Services internal error.
CPFBB9A D	Online value not valid for device &1.
CPFBBA6 E	Server takeover IP address cannot be associated with device subtype &1.
CPIBB10 D	Cluster resource group exit program &1 in library &2 on node &3 failed.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the “Usage Notes” on page 105 above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF2204 E	User profile &1 not found.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C29 E	Object name &1 is not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB0F E	Cluster resource group &1 does not exist in cluster &2.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB2C E	Attributes of exit program &1 in library &2 are not valid.
CPFBB32 E	Attributes of user queue &1 in library &2 not valid.
CPFBB35 E	The user profile name &1 is not valid for this request.
CPFBB38 E	Library name &1 is not allowed for this request.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB43 E	Invalid format name &1 for cluster resource group type &2.
CPFBB44 E	&1 API cannot be called from a cluster resource group exit program.
CPFBB5F E	Number of configuration object entries not valid.
CPFBB60 E	Offset to configuration object array is not valid.

Message ID	Error Message Text
CPFBB61 E	Configuration object &1 specified more than once in configuration object array.
CPFBB63 E	The value specified for the field at offset &1 of configuration object array entry &2 is not valid.
CPFBB6B E	Request not valid for type &1 cluster resource group.
CPFBB6D E	Configuration object &1 not in cluster resource group &2.
CPFBB70 E	API request &1 not compatible with current cluster version.
CPFBBA5 E	Server takeover IP address &1 specified more than once in the configuration object array.
TCP1901 D	Internet address &1 not valid.

API introduced: V5R1

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Create Cluster Resource Group (QcstCreateClusterResourceGroup) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Cluster resource group name	Input	Char(10)
4	Cluster resource group type	Input	Binary(4)
5	Cluster resource group description information	Input	Char(*)
6	Format name	Input	Char(8)
7	Text description	Input	Char(50)
8	Results information	Input	Char(30)
9	Error code	I/O	Char(*)

Service Program: QCSTCRG1
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Create Cluster Resource Group API creates a cluster resource group object. The cluster resource group object identifies a recovery domain. A recovery domain is a set of nodes in the cluster that will play a role in recovery. To change attributes of the cluster resource group use the "Change Cluster Resource Group (QcstChangeClusterResourceGroup) API" on page 86.

This API will do the following for all cluster resource group types:

- Create the cluster resource group object on all nodes in the recovery domain. The cluster resource group may be accessed by a cluster resource group API running on any node in the cluster. The cluster resource group will be owned by the user profile calling this API.
- Provide users a single system image of the cluster resource group object. That is, any changes made to the cluster resource group will be made on all nodes in the recovery domain.
- Call the cluster resource group exit program with an action code of Initialize (1) after the cluster resource group has been created on each node in the recovery domain, if exit program is specified for the cluster resource group. The cluster resource group status will be set to Initialize Pending (540). If the exit program fails, the cluster resource group object is deleted from all nodes in the recovery domain.
- If the exit program is successful, the cluster resource group status is set to Inactive (20). To change the cluster resource group status to Active (10), use the "Start Cluster Resource Group (QcstStartClusterResourceGroup) API" on page 164.
- After the exit program is called, the API verifies the queue used by the Distribute Information API exists if the cluster resource group being created indicates the "Distribute Information

(QcstDistributeInformation) API" on page 127 will be used. The distributed information user queue does not allow pointers within the message content.

- After the exit program is called, the API verifies the failover message queue exists on all recovery domain nodes if one was specified.

This API requires the following for all cluster resource group types:

1. Cluster Resource Services must be active on the node processing the API request.
2. All nodes specified in the recovery domain must be active in the cluster.
3. The cluster resource group exit program must exist on all nodes in the recovery domain if an exit program is specified. It must have the same name and be in the same library on each node.
4. Each node is specified only once in the recovery domain.
5. The cluster resource group name cannot be used by an existing cluster resource group on any node in the cluster.

This API requires the following for resilient application cluster resource groups:

1. For the specified takeover IP address:
   - If Cluster Resource Services configures the takeover IP address, all nodes in the recovery domain must be in the same subnet (network address) and the subnet defined on all nodes in the recovery domain.
   - The takeover IP address must be unique. If Cluster Resource Services is responsible for configuring the takeover IP address, it will be added to each node in the recovery domain.
   - The takeover IP address must not be active on any node in the recovery domain » when cluster resource services creates the application takeover IP address. « .

This API requires the following for resilient device cluster resource groups:

1. Only auxiliary storage pool devices are supported.
2. All nodes in the recovery domain must belong to the same device domain.
3. The configuration objects, such as device descriptions, for the devices specified for the cluster resource group must exist on all nodes in the recovery domain and the resource name specified in a configuration object must be the same on all nodes in the recovery domain.
4. If a data base name is specified in a configuration object, it must be the same on all nodes in the recovery domain.
5. The server takeover IP address must be unique. It can only be associated with a primary auxiliary storage pool.
6. The same configuration object cannot be specified for more than one cluster resource group.
7. Devices attached to the same IOP or high-speed link I/O bridge can be specified for only one cluster resource group. For cross-site mirroring which only has one node at a site, this requirement does not apply.
8. If devices attached to different IOPs or high-speed link I/O bridges are grouped such as for an auxiliary storage pool, all devices for the affected IOPs or high-speed link I/O bridges must be specified in the same cluster resource group. For cross-site mirroring which only has one node at a site, this requirement does not apply.
9. The IOP or high-speed link I/O bridge controlling the devices specified in a cluster resource group must be accessible by all nodes in the cluster resource group's recovery domain or by all nodes within the same site (for cross-site mirroring). This is verified if sufficient hardware configuration has been performed so that all nodes are aware of the new hardware. If hardware configuration is incomplete, this is verified when the Start Cluster Resource Group API is called.
10. If configuration objects are specified and the primary node does not currently own the devices, the API fails with an error message.

11. A cluster resource group may be created with no device entries. Device entries must be added using the "Add Cluster Resource Group Device Entry (QcstAddClusterResourceGroupDev) API" on page 75 before the cluster resource group can be started.
12. If the cluster resource group contains any members of an auxiliary storage pool group, it must contain **all** members before the cluster resource group can be started. All members do not have to be specifed when the cluster resource group is created. Additional members can be added with the Add Cluster Resource Group Device API. If the auxiliary storage pool group exists and clustering can determine the members of the group, a warning message is sent if any members were not specified.
13. If the configuration objects specified are for geographic mirroring, each node in the recovery domain must have a site name and up to 4 data port IP addresses. If a site name is specified, at least one data port IP address must be specified too. The reverse is also true. If one or more data port IP addresses are specified, a site name must be specified too.
14. » If the configuration objects are specified, then the cluster version must be at the appropriate level for the information being specified.«

This API operates in an asynchronous mode. See "Behavior of Cluster Resource Services APIs" on page 221 for more information.

Restriction:

- This API cannot be called from a cluster resource group exit program.
- The cluster resource group name cannot begin with QCST.

Note: For information about the recovery domain, see "Cluster Resource Group APIs" on page 61.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority. This profile is named the calling user profile.

Cluster Resource Group Library Authority
*OBJOPR, *ADD, and *READ

Cluster Resource Group Lock
*EXCL

Exit Program Authority (applies to user profile calling the API and user profile to run the exit program)
*EXECUTE

Exit Program Library Authority (applies to user profile calling the API and user profile to run the exit program)
*EXECUTE

User Profile Authority (applies to user profile to run the exit program)
*USE

Request Information User Queue Authority
*OBJOPR, *ADD

Request Information User Queue Library Authority
*EXECUTE

Request Information User Queue Lock
*EXCLRD

Configuration Object Authority
*USE and *OBJMGT

» *Configuration Object Lock*
*EXCLRD «

Distribute Information User Queue Authority
*OBJOPR, *ADD

Distribute Information User Queue Library Authority
*EXECUTE

Failover Message Queue Authority
**OBJOPR, *ADD*

Failover Message Queue Library Authority
*EXECUTE

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster which will contain the cluster resource group.

Cluster resource group name
INPUT; CHAR(10)

The name of the cluster resource group which is to be created. The cluster resource group name cannot begin with QCST. The cluster resource group object will be created in the QUSRSYS library.

Cluster resource group type
INPUT; BINARY(4)

The type of cluster resource group being created. Valid cluster resource group types are:

1	Data resiliency
2	Application resiliency
3	Device resiliency
» 4	Peer resiliency «

Cluster resource group description information
INPUT; CHAR(*)

Detailed information about the cluster resource group. For more information, see "Data Resiliency (RGDI0100 Format)" on page 111, "Application Resiliency (RGDI0200 Format)" on page 112, "Device Resiliency (RGDI0300 Format)" on page 113, and » Peer Resiliency (RGDI0400 Format) (page "Peer Resiliency (RGDI0400 Format)" on page 114) «.

Format name
INPUT; CHAR(8)

The content and format of the cluster resource group information. The possible values for format name are:

"Data Resiliency (RGDI0100 Format)" on page 111	This format describes the cluster resource group type 1 (data resiliency).
"Application Resiliency (RGDI0200 Format)" on page 112	This format describes the cluster resource group type 2 (application resiliency).

"Device Resiliency (RGDI0300 Format)" on page 113	This format describes the cluster resource group type 3 (device resiliency)
RGDI0400 (page "Peer Resiliency (RGDI0400 Format)" on page 114)	This format describes the cluster resource group type 4 (peer resiliency)

Text description
INPUT; CHAR(50)

This text briefly describes the cluster resource group.

Results information
INPUT; CHAR(30)

This parameter identifies a qualified user queue field and is followed by a reserved field.

Qualified user queue: Completion information is returned to this user queue, which exists on the node from which the API was called, after the function has completed. See the "Usage Notes" on page 121 section of this API for a description of the data that is placed on this queue. This is a 20-character field. The first 10 characters contain the user queue name, and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, *CURLIB are not valid library names. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of the 30-character results information are reserved. Each character in this field must be set to hexadecimal zero.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Data Resiliency (RGDI0100 Format)

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	CHAR(10)	Cluster resource group exit program name
10	A	CHAR(10)	Cluster resource group exit program library name
20	14	CHAR(8)	Cluster resource group exit program format name
28	1C	CHAR(10)	User profile
38	26	CHAR(1)	Additional fields used
39	27	CHAR(1)	Reserved
40	28	CHAR(256)	Exit program data
296	128	BINARY(4)	Offset to recovery domain array
300	12C	BINARY(4)	Number of nodes in recovery domain
304	130	BINARY(4)	Offset to additional fields
308	134	BINARY(4)	Length of additional fields

Offset		Type	Field
Dec	Hex		
*	*	Array (*) of CHAR(*)	Recovery domain array
These fields repeat, in the order listed, for each node in the recovery domain.		CHAR(8)	Node id
		BINARY(4)	Node role
*	*	CHAR(48)	Additional fields
These fields are part of the additional fields structure.		CHAR(10)	Distribute information user queue name
		CHAR(10)	Distribute information user queue library name
		BINARY(4)	Failover wait time
		BINARY(4)	Failover default action
		CHAR(10)	Failover message queue name
		CHAR(10)	Failover message queue library name

Application Resiliency (RGDI0200 Format)

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Cluster resource group exit program name
10	A	CHAR(10)	Cluster resource group exit program library name
20	14	CHAR(8)	Cluster resource group exit program format name
28	1C	CHAR(10)	User profile
38	26	CHAR(1)	Additional fields used
39	27	CHAR(1)	Reserved
40	28	CHAR(256)	Exit program data
296	128	BINARY(4)	Offset to recovery domain array
300	12C	BINARY(4)	Number of nodes in recovery domain
304	130	CHAR(16)	Takeover IP address
320	140	CHAR(10)	Job name
330	14A	CHAR(1)	Configure takeover IP address
331	14B	CHAR(1)	Reserved
332	14C	BINARY(4)	Allow application restart
336	150	BINARY(4)	Number of restarts
340	154	BINARY(4)	Offset to additional fields
344	158	BINARY(4)	Length of additional fields
*	*	Array (*) of CHAR(*)	Recovery domain array
These fields repeat, in the order listed, for each node in the recovery domain.		CHAR(8)	Node id
		BINARY(4)	Node role
*	*	CHAR(49)	Additional fields

Offset		Type	Field
Dec	Hex		
These fields are part of the additional fields structure.		CHAR(10)	Distribute information user queue name
		CHAR(10)	Distribute information user queue library name
		BINARY(4)	Failover wait time
		BINARY(4)	Failover default action
		CHAR(10)	Failover message queue name
		CHAR(10)	Failover message queue library name
		CHAR(1)	Allow active takeover IP address

Device Resiliency (RGDI0300 Format)

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Cluster resource group exit program name
10	A	CHAR(10)	Cluster resource group exit program library name
20	14	CHAR(8)	Cluster resource group exit program format name
28	1C	CHAR(10)	User profile
38	26	CHAR(2)	Reserved
40	28	CHAR(256)	Exit program data
296	128	BINARY(4)	Offset to recovery domain array
300	12C	BINARY(4)	Number of nodes in recovery domain
304	130	BINARY(4)	Length of recovery domain array entry
308	134	BINARY(4)	Offset to configuration object array
312	138	BINARY(4)	Number of configuration object array entries
316	13C	BINARY(4)	Length of configuration object array entry
320	140	BINARY(4)	Offset to additional fields
324	144	BINARY(4)	Length of additional fields
328	148	CHAR(10)	Distribute information user queue name
338	152	CHAR(10)	Distribute information user queue library name
*	*	Array (*) of CHAR(*)	Recovery domain array if Length of recovery domain array entry field is set to non-zero
These fields repeat, in the order listed, for each node in the recovery domain.		CHAR(8)	Node id
		BINARY(4)	Node role
*	*	Array (*) of CHAR(*)	Recovery domain array if Length of recovery domain array entry field is set to zero

Offset		Type	Field
Dec	Hex		
These fields repeat, in the order listed, for each node in the recovery domain.		BINARY(4)	Length of entry in the recovery domain
		CHAR(8)	Node id
		BINARY(4)	Node role
		CHAR(8)	Site name
		BINARY(4)	Offset to data port IP address array
		BINARY(4)	Number of data port IP addresses
		Array(*) of CHAR(16)	Data port IP address
*	*	Array (*) of CHAR(36)	Configuration object array
These fields repeat, in the order listed, for each device entry.		CHAR(10)	Configuration object name
		BINARY(2)	Reserved
		BINARY(4)	Configuration object type
		BINARY(4)	Configuration object online
		CHAR(16)	Server takeover IP address
*	*	CHAR(28)	Additional fields
These fields are part of the additional fields structure.		BINARY(4)	Failover wait time
		BINARY(4)	Failover default action
		CHAR(10)	Failover message queue name
		CHAR(10)	Failover message queue library name

Peer Resiliency (RGDI0400 Format)

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of fixed fields
4	4	BINARY(4)	Offset to recovery domain array
8	8	BINARY(4)	Number of nodes in recovery domain
12	C	BINARY(4)	Length of recovery domain array entry
16	10	CHAR(10)	Cluster resource group exit program name
26	1A	CHAR(10)	Cluster resource group exit program library name
36	24	CHAR(8)	Cluster resource group exit program format name
44	2C	CHAR(10)	User profile
54	36	CHAR(256)	Exit program data
310	136	CHAR(10)	Distribute information user queue name
320	140	CHAR(10)	Distribute information user queue library
330	14A	CHAR(20)	Application id
*	*	Array (*) of CHAR(*)	Recovery domain array

Offset		Type	Field
Dec	Hex		
These fields repeat, in the order listed, for each node in the recovery domain.		CHAR(8)	Node id
		BINARY(4)	Node role «

Field Descriptions

Additional fields. A structure containing optional additional fields.

Additional fields used. A flag to signify whether the additional fields in formats RGDI0100, » RGDI0200, and RGDI0300 « are being used. Possible values are:

0x00 The additional fields are not being used.
0x01 The additional fields are being used.

»

Allow active takeover IP address. Allows a takeover IP address to already be active when it is assigned to an application cluster resource group. This field is only valid when configure takeover IP address field is 0x01. Possible values are:

0 The takeover IP address must not already be active when starting the cluster resource group. This is the default value if the field is not specified.
1 The takeover IP address is allowed to be active prior to starting the cluster resource group but only on the primary node.

«

Allow application restart. Attempt to restart an application if the cluster resource group exit program fails. Possible values are:

0 Do not attempt to restart the application. The cluster resource group exit program is called with an action code of Failover (9).
1 Attempt to restart the application on the same node. The cluster resource group exit program will be called with an action code of Restart (3). If the application cannot be restarted in the specified maximum number of attempts, the cluster resource group exit program will be called with an action code of Failover (9).

»

Application id. This is a string to identify the application supplying the peer cluster resource group. The recommended format is 'vendor-id.name' where vendor-id is an identifier for the vendor creating the cluster resource group and name is the application name. For example, QIBM.ExamplePeer, indicates it is supplied by IBM for the ExamplePeer application. It is not recommended to use QIBM for vendor id name unless the cluster resource group is supplied by IBM.

«

Cluster resource group exit program format name. Indicates which format should be used for the **Information Given To User** parameter on the cluster resource group exit program when it is called. This value must be set to hexadecimal zeroes if an exit program is not specified. Possible values if an exit program is specified are:

EXTP0100 Standard information. This value is allowed for primary-backup model cluster resource group and peer model cluster resource group.

EXTP0200 Additional information in the recovery domain array which contains site name and data port IP addresses on each node. This value is allowed for primary-backup model cluster resource group.

Cluster resource group exit program library name. The name of the library where the exit program exists. The special value *CURLIB or *LIBL may not be used for the library name. QTEMP is not a valid library name. This value must be set to hexadecimal zeroes if an exit program is not specified.

Cluster resource group exit program name. The name of the exit program that is used to handle action codes that are passed to it. The action codes are described in "Cluster Resource Group Exit Program" on page 185.

The cluster resource group exit program cannot be in an independent auxiliary storage pool. Valid special values for this field are:

**NONE* A device cluster resource group may have no cluster resource group exit program. This must be left justified.

Configuration object array. This array identifies the resilient devices that can be switched from one node to another.

Configuration object name. The name of the auxiliary storage pool device description object which can be switched between the nodes in the recovery domain. An auxiliary storage pool device description can be specified in only one cluster resource group.

Configuration object online. Vary the configuration object on and start the server takeover IP address or leave the configuration object varied off and the server takeover IP address inactive when a device is switched from one node to another with the "Initiate Switchover (QcstInitiateSwitchOver) API" on page 135 or when it is failed over to a backup node. This attribute does **not** vary the device on or off and does **not** start or end the server takeover IP address when the cluster resource group is started or ended and when adding a new device entry to the cluster resource group. For secondary auxiliary storage pools, only a value of 2 is valid. If cluster resources cannot determine if this value is correct for a device entry because the auxiliary storage pool is not yet created, any errors will be detected when the cluster resource group is started. A value of 2 cannot be specified for any other device type. Possible values are:

0 Do not vary the configuration object on and do not start the server takeover IP address.
1 Vary the configuration object on and start the server takeover IP address.
2 Perform the same action for a secondary auxiliary storage pool as is specified for the primary.

Configuration object type. This specifies the type of configuration object specified with configuration object name. Possible values are:

1 Device description

Configure takeover IP address. This field identifies who is responsible for configuring (adding and removing) the takeover IP address. This does not affect the starting and ending of the takeover IP address, Cluster Resource Services will perform this function. The following values are valid:

0x00 Cluster Resource Services is responsible for configuring the takeover IP address. The takeover IP address must not exist on any of the nodes in the recovery domain prior to creating cluster resource group. The takeover IP address will be removed when the cluster resource group is deleted.

0x01 User is responsible for configuring the takeover IP address. The takeover IP address must be added on all nodes in the recovery domain except replicates prior to starting the cluster resource group. Using this option it is possible to specify recovery domain nodes in different subnets. See Enabling application switchover across subnets for details.

Data Port IP address. The IP address associated with the recovery domain node. User is responsible for configuring and starting/ending data port IP address. The data port IP address may or may not already exist on the specified node. This field must be represented in dotted decimal format and be a null-terminated string.

Distribute information user queue library name. The name of the library that contains the user queue to receive the distributed information. The library name cannot be *CURLIB, QTEMP, or *LIBL. If the user would like to distribute cluster-wide information through this cluster resource group using the "Distribute Information (QcstDistributeInformation) API" on page 127, then this field must be set. The only way to change the value of this field once the cluster resource group has been created is to delete and recreate the cluster resource group. This field must be set to hexadecimal zeroes if the distribute information user queue name is *NONE.

Distribute information user queue name. The name of the user queue to receive distributed information from the Distribute Information API. If the user would like to distribute cluster-wide information through this cluster resource group using the Distribute Information API, then this field must be set to a value other than *NONE. If this field is set, the specified user queue must exist on all nodes in the recovery domain after the exit program completes.

The queue cannot be in an independent auxiliary storage pool.

The only way to change the value of this field once the cluster resource group has been created is to delete and recreate the cluster resource group. Valid special values for this field are:

**NONE* The Distribute Information API will not be used to distribute information through this cluster resource group.

Exit program data. 256 bytes of data that is passed to the cluster resource group exit program when it is called. This parameter may contain any scalar data except pointers. For example, it can be used to provide state information. This data will be stored with the specified cluster resource group and copied to all nodes in the recovery domain. Pointers in this area will not resolve correctly on all nodes and should not be placed in the data. See "Cluster Resource Group Exit Program" on page 185 for information about the cluster resource group exit program. This value must be set to hexadecimal zeroes if an exit program is not specified.

Failover default action. Should a response to the failover message queue not be received in the failover wait time limit, then this field tells clustering what it should do pertaining to the failover request. This field must be set to 0 if the failover message queue name is *NONE. For format RGDI0100 or RGDI0200, if the current cluster version is 2 and the length of additional fields specified includes the failover default action, this field must be set to 0.

0 Proceed with failover.

1 Do NOT attempt failover.

Failover message queue library name The name of the library that contains the user queue to receive failover messages. The library name cannot be *CURLIB, QTEMP, or *LIBL. If the user would like to receive failover message through this cluster resource group, then this field must be set. This field must be set to hexadecimal zeroes if the failover message queue name is *NONE. For format RGDI0100 or RGDI0200, if the current cluster version is 2 and the length of additional fields specified includes the failover message queue library name, this field must be set to hexadecimal zeroes.

Failover message queue name. The name of the message queue to receive messages dealing with failover. If the user would like to receive notice before a failover occurs, then this field must be set to a value other than *NONE. If this field is set, the specified message queue must exist on all nodes in the recovery domain. The queue cannot be in an independent auxiliary storage pool. For format RGDI0100 or RGDI0200, if the current cluster version is 2 and the length of additional fields specified includes the failover message queue name, this field must be set to hexadecimal zeroes. Valid special values for this field are:

**NONE*	No messages will be sent when a failover occurs through this cluster resource group.

Failover wait time. Number of minutes to wait for a reply to the failover message (CPFBBAB) that was enqueued on the failover message queue. This field must be set to 0 if the failover message queue name is *NONE. This field cannot be set to 0 if a failover message queue is specified. For format RGDI0100 or RGDI0200, if the current cluster version is 2 and the length of additional fields specified includes the failover wait time, this field must be set to 0. Valid values are:

-1	Wait forever until a response is given to the failover inquiry message.
0	Failover proceeds without user intervention. Acts the same as V5R1 and prior.
>=1	Number of minutes to wait for a response to the failover inquiry message. If no response is received in the specified number of minutes, the failover default action field will be looked at to decide how to proceed.

Job name. The name given the batch job that is submitted. This is the job that calls the cluster resource group exit program. Valid special values for this field are:

**JOBD*	The job name in the job description for the specified user profile will be used. This must be left justified.

Length of additional fields. The length in bytes of additional fields. In formats RGDI0100 and RGDI0200, this field is ignored if the additional fields used flag is not set to 1. If the additional fields used flag is 1 in formats RGDI0100 and RGDI0200, the value of this field must be equal to 20, » 48 or 49. « In format RGDI0300, if the cluster version is less than 3, the value of this field must be 0. If the cluster version is 3, the value of this field must be equal to 0 or 28.

Length of configuration object array entry. The length of an entry in the configuration object array. This field must be set to 0 if the number of entries in the configuration object array field has a value of 0. If the number of entries has a value greater than 0, it must be set to the length of a single entry.

Length of entry in the recovery domain. The length of an entry in the recovery domain array. This field is used if each entry may have a different length.

» Length of fixed fields. The length of the fixed fields in the format description. For RGDI0400 this value must be 350. «

Length of recovery domain array entry. The length of an entry in the recovery domain array. For formats other than RGDI0300, this field must be set to 12. For format RGDI0300, this field can be set to either 12 or zero. If zero, then the length of entry in the recovery domain field is used.

Node id. A unique string of characters that identifies a node that is participating in the recovery domain of the specified cluster resource group. The node specified must be active in the cluster, and it must be unique in the recovery domain of the specified cluster resource group.

Node role. The role the node has in the recovery domain. A role must be defined for each node in the recovery domain. For primary-backup model cluster resource groups a node can have one of three roles: primary, backup, or replicate. Only one node can be designated as the primary. This node will become an active access point when the cluster resource group is started. Backup nodes are assigned a backup order. One indicates the first backup, two the second backup, and so on. Backup nodes are avaliable to become an active access point. Replicates are not ordered and cannot become an access point for the cluster resource unless the "Change Cluster Resource Group (QcstChangeClusterResourceGroup) API" on page 86 is used to change its role from replicate to either a backup or primary.
For peer model cluster resource groups a node can have one of two roles: peer or replicate. Any number of nodes can be designated as the peer or replicate. Peer nodes are not ordered and can become an active access point for the cluster resources. Replicates are not ordered and cannot become an active access point for the cluster resource unless the "Change Cluster Resource Group (QcstChangeClusterResourceGroup) API" on page 86 is used to change its role from replicate to peer.
The following summarizes the valid values for this field:

0	Primary node. Only one node can have this value.
>=1	Backup node. The backup order is designated by increasing value. The values need not be consecutive. No two backup nodes can have the same value. At the completion of the API, Cluster Resource Services will sequence the backups using consecutive numbers starting with 1.
-1	Replicate node. All replicates have this value.
-4	Peer node. All peers have this value.

Number of configuration object array entries. The number of entries in the configuration object array. This value can be 0 if no configuration object entries are to be added when the cluster resource group is initially created. At least one configuration object entry must be added before the "Start Cluster Resource Group (QcstStartClusterResourceGroup) API" on page 164 API is called. A cluster resource group can have a maximum of 256 configuration object entries. If no configuration objects specified this field must be hexadecimal zero.

Number of data port IP addresses. The number of data port IP addresses associated with the recovery domain node. A node can have up to 4 data port IP addresses. If the current cluster version is 3 and the length of recovery domain array specified includes this field, it must be set to zero.

Number of nodes in the recovery domain. The number of nodes in the recovery domain array. This should equal the number of backup nodes plus the number of replicate nodes plus one (for the primary node). This must be greater than or equal to one and equal the number of nodes in the recovery domain.

Number of restarts. Number of times an application can be restarted on the same node before a failover occurs. Maximum number of restarts is 3. Every time the cluster resource group exit program is run, the number of restarts is reset to 0 and works up to the maximum value specified.

Offset to additional fields. The byte offset from the beginning of this parameter to additional fields. In formats RGDI0100 and RGDI0200, this field will be ignored unless the additional fields used field is set to 1. In format RGDI0300, if the cluster version is less than 3, the value of this field must be 0.

Offset to configuration object array. The byte offset from the beginning of this parameter to the Configuration object array field. This field must be set to 0 if the number of entries in the configuration object array field has a value of 0.

Offset to data port IP address array. The byte offset from the beginning of this parameter to the first data port IP address. If the current cluster version is 3 and the length of recovery domain array specified includes this field, it must be set to zero. This field is ignored if the number of data port IP addresses field is set to zero.

Offset to recovery domain array. The byte offset from the beginning of this parameter to the Recovery domain array field.

Recovery domain array. This array identifies the nodes that compose the recovery domain. A role must be defined for each node in the recovery domain. Nodes in the recovery domain must be unique. See the node role field for more information on primary, backup, replicate » and peer. «

An example » for a primary-backup model cluster resource group «: A cluster resource group has four nodes: NodeA, NodeB, NodeC and NodeD. NodeA is the primary. There are two backup nodes: NodeB and NodeD. NodeD is the first backup and NodeB is the second backup. There is one replicate: NodeC.

```
Node Role

----- ----

NodeA 0 <-- primary

NodeD 1 <-- backup #1

NodeB 2 <-- backup #2

NodeC -1 <-- replicate
```

The nodes do not have to be arranged in any particular order in the array. They could be in the array as listed below and have the same result.

```
Node Role

----- ----

NodeB 2 <-- backup #2

NodeA 0 <-- primary

NodeC -1 <-- replicate

NodeD 1 <-- backup #1
```

Reserved. Must contain hexadecimal zeroes.

Server takeover IP address. This is a takeover IP address for servers associated with the relational database name in the device description for an auxiliary storage pool. This field is optional and can only be specified for a primary auxiliary storage pool. If specified, the address must be represented in dotted decimal format and be a null-terminated string. The specified address must exist on all nodes in the recovery domain if the cluster resource group is active. If not specified, or for a secondary and UDFS auxiliary storage pool, this field must be set to *NONE and be left justified. If the current cluster version is 2 and the length of configuration object array entry specified includes the server takeover IP address, this field must be set to hexadecimal zeroes. Valid special values for this field are:

*NONE	There is no server takeover IP address associated with the relational database name in the device description for an auxiliary storage pool.

Site name. The name of the site associated with the recovery domain node. If the current cluster version is 3 and the length of recovery domain array specified includes this field, this field must be set to hexadecimal zeroes. Valid special values for this field are:

*NONE The node in the recovery domain is not associated with a site name. This must be left justified.

Takeover IP address. This is the floating IP address that is to be associated with the application. This field must be represented in dotted decimal format and be a null-terminated string. The Cluster Resource Services will create this IP address on every system in the recovery domain if the Configure takeover IP address is 0x00. If the IP address already exists, then this API will fail.

User profile. The name of the user profile under which the exit program should process. The user profile must exist on all nodes in the recovery domain. This field must be set to hexadecimal zeroes if the cluster resource group does not have an exit program. The following user profiles are not valid:

- QDBSHR
- QDOC
- QDFTOWN
- QRJE
- QLPAUTO
- QLPINSTALL
- QSECOFR
- QSPL
- QSYS
- QTSTRQS

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See “Cluster APIs Use of User Queues” on page 221 and “Using Results Information” on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Error Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF18BA D	Error occurred with subsystem.
CPF2204 D	User profile &1 not found.
CPF2108 D	Object not added to library.
CPF2112 D	Object &1 in &2 type *&3 already exists.
CPF2113 D	Cannot allocate library &1.
CPF2182 D	Not authorized to library &1.
CPF3C29 D	Object name &1 is not valid.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPF9801 D	Object &2 in library &3 not found.
CPF9802 D	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 D	Object &2 in library &3 damaged.
CPF9810 D	Library &1 not found.
CPF9820 D	Not authorized to use library &1.
CPF9830 D	Cannot assign library &1.
CPF9838 D	User profile storage limit exceeded.
CPF9870 D	Object &2 type *&5 already exists in library &3.
CPFBB02 D	Cluster &1 does not exist.

Message ID	Error Message Text
CPFBB0A D	Node &1 is not active in cluster &2.
CPFBB0B D	Request using takeover IP address &1 failed.
CPFBB17 D	&1 API cannot be processed in cluster &2.
CPFBB27 D	A primary node was not specified for the recovery domain.
CPFBB28 D	Cluster node &1 and cluster node &2 have the same node role value &3.
CPFBB29 D	Node role value &1 not valid.
CPFBB2C D	Attributes of exit program &1 in library &2 are not valid.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB2E D	Job submission failed for cluster resource group &1 in cluster &2.
CPFBB30 D	Takeover IP address &1 is not part of the TCP/IP subnetwork.
CPFBB32 D	Attributes of user queue &1 in library &2 are not valid.
CPFBB34 D	Cluster resource group &1 already exists in cluster &2.
CPFBB35 D	The user profile name &1 is not valid for this request.
CPFBB38 D	Library name &1 is not allowed for this request.
CPFBB39 D	Current user does not have IOSYSCFG special authority.
CPFBB46 D	Cluster Resource Services internal error.
CPFBB47 D	Cluster Resource Services ended abnormally.
CPFBB48 D	Cluster Resource Services error detected.
CPFBB51 D	IP address &1 already in use by the cluster &3.
CPFBB5A D	All recovery domain nodes not in same device domain.
CPFBB5B D	Resource name &1 incorrect for configuration object &2 on node &3.
CPFBB5C D	Configuration object &1 already in cluster resource group &2.
CPFBB5D D	Other related devices already in cluster resource group &1.
CPFBB60 D	Cluster message not received from cluster node &3.
CPFBB64 D	Configuration object &1 not valid device type.
CPFBB66 D	Request failed for device cluster resource group &3.
CPFBB7A D	Primary node &1 in cluster resource group &2 not current owner of specified devices.
CPFBB7B D	Device type incorrect for configuration object &1 on node &2.
CPFBB7C D	Resource name &1 already used by configuration object &2 in cluster resource group &4.
CPFBB7D D	Configuration object &1 already in cluster resource group &2.
CPFBB7E D	Resource name &1 already in cluster resource group &2.
CPFBB7F D	Too many I/O processors or high-speed link I/O bridges specified for cluster resource group &1.
CPFBB80 D	Request failed for device cluster resource group &3.
CPFBB84 D	Device domain entry for node &1 being removed.
CPFBB90 D	Request failed for device cluster resource group &3.
CPFBB98 D	Hardware resource &1 not switchable.
CPFBB99 D	Request failed for device cluster resource group &3.
CPFBB9A D	Online value not valid for device &1.
CPFBB9B D	Auxiliary storage pool group member &1 not specified.
CPFBB9D D	Device &1 not compatible with current cluster version.
CPFBB9E D	Data base name &1 not correct for configuration object &2 on node &3.
CPFBBA6 D	Server takeover IP address cannot be associated with device subtype &1.
CPFBBA8 D	Site name &1 specified for node &2 not allowed.
CPFBBA9 D	Data port IP address &1 specified for node &2 not allowed.
CPIBB10 D	Cluster resource group exit program &1 in library &2 on node &3 failed.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the “Usage Notes” on page 121 above.

Message ID	Error Message Text
CPF2112 E	Object &1 in &2 type *&3 already exist.
CPF2113 E	Cannot allocate library &1.

Message ID	Error Message Text
CPF2204 E	User profile &1 not found.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C29 E	Object name &1 is not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3C4B E	Value not valid for field &1.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9838 E	User profile storage limit exceeded.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB09 E	Cluster node &1 does not exist in cluster &2.
CPFBB0A E	Cluster node &1 in cluster &2 is not active.
CPFBB0E E	Cluster resource group type &1 not valid.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB27 E	No primary node specified in recovery domain.
CPFBB28 E	Two or more backup nodes have the same node role value &1.
CPFBB29 E	Node role value &1 not valid.
CPFBB2C E	Attributes of exit program &1 in library &2 are not valid.
CPFBB31 E	Value &1 specified for number of restarts not valid.
CPFBB32 E	Attributes of user queue &1 in library &2 are not valid.
CPFBB33 E	Cluster node &1 already exists in the recovery domain for cluster resource group &4
CPFBB34 E	Cluster resource group &1 already exist in cluster &2.
CPFBB35 E	User profile name not valid for this request.
CPFBB36 E	Number of nodes in recovery domain is not valid.
CPFBB37 E	Offset to recovery domain is not valid.
CPFBB38 E	Library name &1 not allowed for this request.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB40 E	The value &1 specified for the allow application restart parameter is not valid.
CPFBB43 E	Format name &1 not valid for cluster resource group type &2.
CPFBB44 E	&1 API cannot be called from a cluster resource group exit program.
CPFBB51 E	IP address &1 already in use by the cluster &3.
CPFBB5A E	Device domain for recovery domain nodes not correct.
CPFBB5F E	Field value within structure is not valid.
CPFBB61 E	Configuration object &1 specified more than once in configuration object array.
CPFBB62 E	Exit program name *NONE not valid.
CPFBB63 E	Configuration object &1 not valid type.
CPFBB64 E	Configuration object &1 not valid device type.
CPFBB70 E	API request &1 not compatible with current cluster version.
CPFBBA2 E	Value &1 specified for failover wait time is not valid.
CPFBBA3 E	Value &1 specified for failover default action is not valid.
CPFBBA4 E	Field value within additional fields structure is not valid.
CPFBBA5 E	Server takeover IP address &1 specified more than once in the configuration object array.
CPFBBA7 E	Site name and data port IP address not match.
CPFBBA8 E	Site name &1 specified for node &2 not allowed.
CPFBBA9 E	Data port IP address &1 specified for node &2 not allowed.
CPFBBAC E	The offset to the data port IP address array for node &1 is not valid.

Message ID	Error Message Text
CPFBBAD E	The number of data port IP addresses specified for node &1 is not valid.
TCP1901 E	Internet address &1 not valid.

API introduced: V4R4

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Delete Cluster Resource Group (QcstDeleteClusterResourceGroup) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Cluster resource group name	Input	Char(10)
4	Results information	Input	Char(30)
5	Error code	I/O	Char(*)

Service Program: QCSTCRG1
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Delete Cluster Resource Group API deletes a cluster resource group from all nodes in the recovery domain.

The cluster resource group object is marked for deletion and is deleted on each active cluster node. The cluster resource group object will be deleted on other nodes in the cluster when they become active. If an exit program is specified for the cluster resource group, the cluster resource group exit program is called on each active node in the recovery domain with an action code of Verification Phase (5) and action code dependent data of Delete (12). This gives the exit program the ability to verify the exit program. If this call returns with a failure, the delete request will not proceed. If this call returns with success, the exit program will be called with Delete (7). The cluster resource group status is set to Delete Pending (550). The cluster resource group will be deleted even if the exit program fails. This API will never call the cluster resource group exit program with an action code of Undo (15).

The Delete Cluster Resource Group (DLTCRG) command can be used to delete a cluster resource group object on a system that does not have Cluster Resource Services active.

Deleting a device cluster resource group will not change the ownership of devices. The devices remain on whatever nodes owns them at the time of the delete.

If Cluster Resource Services configured the takeover IP address for an application cluster resource group and the IP interface is not active, the takeover IP address will be removed. If Cluster Resource Services finds that the takeover IP address is active, the API will fail with an error message.

This API requires:

1. Cluster Resource Services active on the node processing the request.
2. Cluster resource group status must not be Active (10).

This API operates in an asynchronous mode. See "Behavior of Cluster Resource Services APIs" on page 221 for more information.

Restriction: This API cannot be called from a cluster resource group exit program.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

Cluster Resource Group Authority
*OBJEXIST, *USE

Cluster Resource Group Library Authority
*EXECUTE

Cluster Resource Group Lock
*EXCL

Exit Program Authority » (applies to user profile calling the API and user profile to run the exit program) «
*EXECUTE

Exit Program Library Authority » (applies to user profile calling the API and user profile to run the exit program) «
*EXECUTE

User Profile Authority » (applies to user profile to run the exit program) «
*USE

Request Information User Queue Authority
*OBJOPR, *ADD

Request Information User Queue Library Authority
*EXECUTE

Request Information User Queue Lock
*EXCLRD

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster containing the cluster resource group.

Cluster resource group name
INPUT; CHAR(10)

The name of the cluster resource group.

Results information
INPUT; CHAR(30)

This parameter identifies a qualified user queue field and is followed by a reserved field.

Qualified user queue: Completion information is returned to this user queue, which exists on the node from which the API was called, after the function has completed on all active nodes in the cluster. See the “Usage Notes” on page 126 section of this API for a description of the data that is placed on this queue. This is a 20-character field. The first 10 characters contain the user queue name, and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, *CURLIB are not valid library names. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of the 30-character results information are reserved. Each character in this field must be set to hexadecimal zero.

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See "Cluster APIs Use of User Queues" on page 221 and "Using Results Information" on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Error Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF18BA D	Error occurred with subsystem.
CPF2113 E	Cannot allocate library &1.
CPF2204 D	User profile &1 not found.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPF9801 D	Object &2 in library &3 not found
CPF9802 D	Not authorized to object &2 in &3.
CPF9803 D	Cannot allocate object &2 in library &3.
CPF9804 D	Object &2 in library &3 damaged.
CPF9810 D	Library &1 not found
CPFBB02 D	Cluster &1 does not exist.
CPFBB0F D	Cluster resource group &1 does not exist in cluster &2.
CPFBB18 D	Request &1 not allowed for cluster resource group &2.
CPFBB2C D	Attributes of exit program &1 in library &2 are not valid.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB2E D	Job submission failed for cluster resource group &1 in cluster &2.
CPFBB39 D	Current user does not have IOSYSCFG special authority.
CPFBB46 D	Cluster resource service internal error.
CPFBB47 D	Cluster Resource Services ended abnormally.
CPIBB10 D	Cluster resource group exit program &1 in library &2 on node &3 failed.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the "Usage Notes" above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF2204 E	User profile &1 not found.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C1E E	Required parameter &1 omitted.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 D	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.

Message ID	Error Message Text
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB0F E	Cluster resource group &1 does not exist in cluster &2.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB2C E	Attributes of exit program &1 in library &2 are not valid.
CPFBB32 E	Attributes of user queue &1 in library &2 are not valid.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB44 E	&1 API cannot be called from a cluster resource group exit program.

API introduced: V4R4

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Distribute Information (QcstDistributeInformation) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Cluster resource group name	Input	Char(10)
4	Distribute information	Input	Char(*)
5	Format name	Input	Char(8)
6	Results information	Input	Char(30)
7	Error code	I/O	Char(*)

Service Program: QCSTCRG2
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Distribute Information (QcstDistributeInformation) API is used to deliver information from a node in the recovery domain to other nodes in the recovery domain. The information will be delivered to the active nodes in the recovery domain for that cluster resource group. The number of nodes field in format RGDD0100 is used to send the information to only one node or all nodes.

The amount of information that can be sent between nodes with this API is between 1 and 62,000 bytes in length. The API does not encrypt the message data before sending it.

When the information is delivered to a node, it will be placed on the specified user queue. If the user queue is keyed, the key length must be greater than zero and the key must be at the beginning of the data. If the key length is 0, the queue will be assumed to be non-keyed. The user queue cannot be in an independent auxiliary storage pool. The results information queue will indicate the success or failure of the distribution for each active node in the recovery domain. For each failed delivery, a diagnostic message will be placed on the results information queue. When the API completes, a completion message is placed on the results information queue. » If the message type is "First-In,First-Out" then no completion message will be placed on the results information queue. «

This API can only be run on a node that is an active member in the recovery domain of a cluster resource group. For example, if nodes A, B, and C are active and nodes A and B make up the recovery domain for CRGA, the Distribute Information API for CRGA can only be run on nodes A or B.

Message delivery is determined by the value specified for message delivery type field. If you want all nodes in the recovery domain to receive the message in the same sequence, specify a 0. If you want to just send the message without guarantee of delivery, specify a 1. Messages are ordered within the same delivery type from the same sender.

This API requires the qualified distribute information user queue name field to be specified when the cluster resource group is created, otherwise this API will not be allowed to execute.

This API will:

1. Distribute the data to each node in the recovery domain with a status of Active (10) or to a single node if the number of nodes field in RGDD0100 is set to 1.
2. Enqueue the data on the user queue specified.
3. Work when the cluster is in a partition state. When the cluster is partitioned, the message is sent only to those nodes that are in the same partition as the node where the API was called.

This API requires:

1. Cluster Resource Services active on the node processing the request.
2. Existence of the specified user queue on all nodes in the array of node ids specified on this API.
3. The key at the beginning of the data, if using a keyed user queue.
4. The node processing the request must be in the recovery domain of the cluster resource group, otherwise the API is not valid.

This API operates in asynchronous mode. See "Behavior of Cluster Resource Services APIs" on page 221 for more information. This API may be called from a cluster resource group exit program. The cluster resource group exit program will not be called when this API is run.

Restriction: This API should not be called from a cluster resource group exit program when a node is joining this cluster resource group, a node is being added to the recovery domain of this cluster resource group, or this cluster resource group is being created. The reason for this is that the node that is joining or being added will not get the information that the Distribute Information API sent.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

User Queue Authority
: *OBJOPR and *ADD

User Queue Library Authority
: *EXECUTE

User Queue Lock
: *EXCLRD

Request Information User Queue Authority
: *OBJOPER, *ADD

Request Information User Queue Library Authority
: *EXECUTE

Request Information User Queue Lock
: *EXCLRD

User Profile Authority
: *USE

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

If the message type is »"First-In,First-Out"« then the request handle field will be blank.

Cluster name
INPUT; CHAR(10)

The name of the cluster to which the cluster resource group belongs.

Cluster resource group name
INPUT; CHAR(10)

This name of the cluster resource group.

Distribute information
INPUT; CHAR(*))

The detailed information about the message that is be distributed. For more information, see "Distribute Information (RGDD0100 Format)."

Format name
INPUT;CHAR(8)

The content and format of the distribute information parameter. The possible format names are:

"Distribute Information (RGDD0100 Format)"	Distribute information

Results information
INPUT; CHAR(30)

A library qualified user queue name followed by a reserved field.

Library qualified user queue: A user queue, which exists on the node from which the API was called, that receives results information after the function has completed. See the "Usage Notes" on page 130 section of this API for a description of the data that is placed on this queue. This is a 20 character field. The first 10 characters contain the user queue name and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, and *CURLIB are not valid for the library name. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of results information are reserved and must be set to hexadecimal zero.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code parameter.

Distribute Information (RGDD0100 Format)

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Message delivery type
4	4	BINARY(4)	Length of message key
8	8	BINARY(4)	Offset to message

Offset		Type	Field
Dec	Hex		
12	C	BINARY(4)	Length of message
16	10	BINARY(4)	Offset to array of node ids
20	14	BINARY(4)	Number of node ids
24	18	CHAR(*)	Message
*	*	Array of CHAR(8)	Node id

Field Descriptions

Length of message. The length of the data that is to be distributed. This includes the key length. This value must be between 1 and 62,000.

Length of message key. The length of the key used to enqueue the message on a keyed user queue. The key must be at the start of the distributed message. Possible values are:

0 Non-keyed user queue.
>=1 Size of the key.

Message. The message to be distributed to the nodes in the recovery domain of the cluster resource group or node id array. Since pointers will not resolve correctly on other nodes, pointers should not be included in the message.

Message delivery type. The method used to deliver the message. Possible values are:

0 Total ordering. Guaranteed delivery of totally ordered message and that receipt of message on all active nodes in the recovery domain.
1 First-In, First Out (FIFO). Non guaranteed delivery. Message delivered FIFO and no receipt/acknowledgment required.

Node id. Name of the node to receive the message. If number of node ids is not 1, this field is ignored. If the number of node ids is 1, this value must contain the name of an active node in the recovery domain.

Number of node ids. The number of nodes in the array of node ids. If message delivery type is 0, the number of node ids must be 0. If message delivery type is 1, this field must contain 1 or -1. Possible values are:

-1 The message will be sent point to point to all active nodes in the recovery domain.
0 Total ordering, no node ids can be specified. Message processed on all nodes in the recovery domain at the same time.
1 Message deliver is First-In, First Out(FIFO). Only one node id can be specified.

Offset to array of node ids. The byte offset from the beginning of this table to the first node id. If number of node ids is -1 or 0, this field must be zero.

Offset to message. The byte offset from the beginning of this table to the message.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See “Cluster APIs Use of User Queues” on page 221 and “Using Results Information” on page 223

for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF2113 D	Cannot allocate library &1.
CPF2204 D	User profile &1 not found.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPF9801 D	Object &2 in library &3 not found.
CPF9802 D	Not authorized to object &2 in &3.
CPF9803 D	Cannot allocate object &2 in library &3.
CPF9804 D	Object &2 in library &3 damaged.
CPF9810 D	Library &1 not found.
CPFBB0A D	Cluster node &1 in cluster &2 not active.
CPFBB0F D	Cluster resource group &1 does not exist in cluster &2.
CPFBB18 D	Request &1 is not allowed for cluster resource group &2.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB32 D	Attributes of user queue &1 in library &2 are not valid.
CPFBB38 D	Library name &1 is not allowed for this request.
CPFBB46 D	Cluster Resource Services internal error.
CPFBB47 D	Cluster Resource Services ended abnormally.
CPFBB5F D	Field value within structure is not valid.
CPFBB8F D	Enqueue on distribute information queue &1 in library &2 failed.
CPIBB0B I	Distribute information message delivered to &1.
CPIBB0C I	Distribute information message could not be delivered to &1.
CPIBB0D I	No attempt made to deliver distribute information message to node &1.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the “Usage Notes” on page 130 above.

Message ID	Error Message Text
CPF2204 E	User profile &1 not found.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C1E E	Required parameter &1 omitted.
CPF3C29 E	Object name &1 is not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB0F E	Cluster resource group &1 does not exist in cluster &2.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB32 E	Attributes of user queue &1 in library &2 not valid.
CPFBB38 E	Library name &1 is not allowed for this request.
CPFBB46 E	Cluster Resource Services internal error.
CPFBB5F E	Field value within structure is not valid.

Message ID	Error Message Text
CPFBB70 E	API request &1 not compatible with current cluster version.
CPFBB88 E	Node &1 not in recovery domain for Cluster Resource Group &2.
CPFBB8C E	&2 API will not work against cluster resource group &1.

API introduced: V5R1

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

End Cluster Resource Group (QcstEndClusterResourceGroup) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Cluster resource group name	Input	Char(10)
4	Exit program data	Input	Char(256)
5	Results information	Input	Char(30)
6	Error code	I/O	Char(*)

Service Program: QCSTCRG2
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The End Cluster Resource Group (QcstEndClusterResourceGroup) API disables resiliency of the specified cluster resource group. If an exit program is specified for the cluster resource group, the cluster resource group exit program is called with an action code of End (4) on each active node in the recovery domain.

When the exit program is called, the cluster resource group status is set to End Cluster Resource Group Pending (550). Successful completion of the exit program sets the cluster resource group status Inactive (20). After the exit program completes successful, for an application cluster resource group:

- The current exit program job will be cancelled with the *IMMED option
- The takeover IP interface for the cluster resource group will be ended.

If the exit program fails and the original state of the cluster resource group cannot be recovered, the cluster resource group status is set to Indoubt (30).

Ending a device cluster resource group will not change the ownership of devices. The devices remain on whatever nodes owns them at the time the API is run. Also, the devices are not varied off when the cluster resource group is ended.

» Ending a peer cluster resource group will end the access point for cluster resources on all nodes defined as a peer node. «

This API requires:

1. Cluster Resource Services started on the node processing the request.
2. Cluster resource group status of Active (10) or Indoubt (30).

This API operates in an asynchronous mode. See "Behavior of Cluster Resource Services APIs" on page 221 for more information.

Restriction: This API cannot be called from a cluster resource group exit program.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

Cluster Resource Group Authority
*CHANGE

Cluster Resource Group Library Authority
*EXECUTE

Cluster Resource Group Lock
*EXCL

Exit Program Authority » (applies to user profile calling the API and user profile to run the exit program) «
*EXECUTE

Exit Program Library Authority » (applies to user profile calling the API and user profile to run the exit program) «
*EXECUTE

User Profile Authority » (applies to user profile to run the exit program) «
*USE

Request Information User Queue Authority
*OBJOPR, *ADD

Request Information User Queue Library Authority
*EXECUTE

Request Information User Queue Lock
*EXCLRD

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster containing the cluster resource group.

Cluster resource group name
INPUT; CHAR(10)

The name of the cluster resource group which will be ended.

Exit program data
INPUT; CHAR(256)

256 bytes of data that is passed to the cluster resource group exit program when it is called. This parameter may contain any scalar data except pointers. For example, it can be used to provide state information. This data will be stored with the specified cluster resource group and copied to all nodes in the recovery domain. Pointers in this area will not resolve correctly on all nodes and should not be placed in the data. See “Cluster Resource Group Exit Program” on page 185 for information about the cluster resource group exit program. The data specified will replace the existing exit program data stored with the cluster resource group. If blanks are specified, then the exit program data stored with the cluster resource group will be cleared. This parameter must be set to *SAME if no exit program is specified for the cluster resource group. The following special value can be used:

*SAME The exit program data stored with the cluster resource group specified will be passed to the exit program. This must be left justified.

Results information
INPUT; CHAR(30)

This parameter identifies a qualified user queue field and is followed by a reserved field.

Qualified user queue: Completion information is returned to this user queue, which exists on the node from which the API was called, after the function has completed. See the "Usage Notes" section of this API for a description of the data that is placed on this queue. This is a 20-character field. The first 10 characters contain the user queue name, and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, *CURLIB are not valid library names. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of the 30-character results information are reserved. Each character in this field must be set to hexadecimal zero.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See "Cluster APIs Use of User Queues" on page 221 and "Using Results Information" on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Error Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF18BA D	Error occurred with subsystem.
CPF2113 E	Cannot allocate library &1.
CPF2204 D	User profile &1 not found.
CPF9801 D	Object &2 in library &3 not found.
CPF9802 D	Not authorized to object &2 in &3.
CPF9803 D	Cannot allocate object &2 in library &3.
CPF9804 D	Object &2 in library &3 damaged.
CPF9810 D	Library &1 not found.
CPFBB0F D	Cluster resource group &1 does not exist in cluster &2.
CPFBB17 D	&1 API cannot be processed in cluster &2.
CPFBB18 D	Request &1 not allowed for cluster resource group &2.
CPFBB2C D	Attributes of exit program &1 in library &2 are not valid.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB2E D	Job submission failed for cluster resource group &1 in cluster &2.
CPFBB32 D	Attributes of user queue &1 are not valid.
CPFBB47 D	Cluster Resource Services ended abnormally.
CPFBB48 D	Cluster Resource Services error detected.
CPFBB6E E	Exit program data cannot be specified.
CPIBB10 D	Cluster resource group exit program &1 in library &2 on node &3 failed.
TCP1B61 D	Unable to determine if &1 interface ended.&2 successful (&3 %).
TCP1B62 D	Cannot determine if &1 interface ended.

Message ID	Error Message Text
TCP1B65 D	&2 interface not ended. Reason &1.
TCP1B72 D	&1 interface not ended. &1 interface is not active.
TCP1B73 D	&1 interface not ended. &1 interface not defined in TCP/IP configuration.
TCP1B74 D	&1 interface not ended. Line description &2 not found.
TCP1B85 D	&1 interface not ended.
TCP9999 D	Internal system error in program &1.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the "Usage Notes" on page 134 above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF2204 E	User profile &1 not found.
CPF24B4 E	Severe error while addressing parameter list.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF3C1E E	Required parameter &1 omitted.
CPF3C39 E	Value for reserved field not valid.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPFBB02 E	Cluster &1 does not exist.
CPFBB0F E	Cluster resource group &1 does not exist in cluster &2.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB2C E	Attributes of exit program &1 in library &2 are not valid.
CPFBB32 E	Attributes of user queue &1 are not valid.
CPFBB38 E	Library name &1 not allowed for this request.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB44 E	&1 API cannot be called from a cluster resource group exit program.
CPFBB46 E	Cluster Resource Services internal error.
CPFBB6E E	Exit program data cannot be specified.

API introduced: V4R4

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Initiate Switchover (QcstInitiateSwitchOver) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Cluster resource group name	Input	Char(10)
4	Exit program data	Input	Char(256)
5	Results information	Input	Char(30)
6	Error code	I/O	Char(*)

Service Program: QCSTCRG2
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Initiate Switchover (QcstInitiateSwitchOver) API changes the current roles of nodes in the recovery domain of a cluster resource group:

- The current primary node is assigned the role of last active backup.
- The current first backup is assigned the role of primary.

If a backup node does not exist in the recovery domain, the switchover will fail. If the first backup is not the desired primary, first use the "Change Cluster Resource Group (QcstChangeClusterResourceGroup) API" on page 86 API to arrange the backup nodes in recovery domain to the desired order.

This API will do the following for all cluster resource group types:

1. Set the cluster resource group status Switchover Pending (570).
2. Call the cluster resource group exit program on all active nodes in the recovery domain with an action code of Switchover (10), if an exit program is specified for the cluster resource group.
3. Set the cluster resource group status to Active (10) if the exit program completes successfully.
4. Set the cluster resource group status to Indoubt (30) if the exit program is unsuccessful and the original state of the cluster resource group cannot be recovered.

This API will do the following for resilient application cluster resource groups:

1. Cancel the cluster resource group exit program job with a Cancel Job Immediate on the current primary.
2. End the takeover IP interface on the current primary.
3. Start the takeover IP interface on the new primary.
4. Start the cluster resource group exit program on the new primary.

   Note: The application and exit program code should provide cancel handlers to clean up the job if it is cancelled.
5. Set the cluster resource group status to Active (10) if the takeover IP address and the cluster resource group exit program job are started.
6. Set the cluster resource group status to Indoubt (30) if either the takeover IP address or the cluster resource group exit program job are not started.

This API will do the following for resilient device cluster resource groups:

1. The configuration objects must exist on all active nodes in the recovery domain and the resource names in the configuration objects must be the same on all active nodes.
2. The current primary node must own the IOPs or high-speed link I/O bridges for the devices configured in the cluster resource group.
3. The new primary node must be able to access the IOPs or high-speed link I/O bridges for the devices configured in the cluster resource group. This requirement does not apply to cross-site mirroring if the new primary node is at a different site than the current primary node.
4. For geographic mirroring, if the new primary node is at a different site than the current primary node, a role swap of auxiliary storage pools will occur where a production copy on the current primary node becomes a mirror copy, and a mirror copy on the new primary node becomes a production copy. If there is another active backup node at the same site as the current primary node, the auxiliary storage pools are moved from the current primary node to that site backup node.
5. On the current primary node if the cluster resource group is active, the configuration objects specified in the cluster resource group are varied off and the server takeover IP addresses are ended. The devices are moved to the new primary or the role of the auxiliary storage pools are swapped (for geographic mirroring where the new primary node is at a different site than the current primary node), before the exit program is called on the current primary. If any of the devices in the cluster resource group are a primary auxiliary storage pool, all members of the auxiliary storage pool group will be varied off. Before varying the devices off, cluster resource services will attempt to end all jobs which are using auxiliary storage pools configured in the cluster resource group. There are some

system server jobs which will not be cancelled. If those server jobs are performing long running operations against data on an auxiliary storage pool, the devices may not vary off and the switchover will fail.

6. For the configuration objects specified in the cluster resource group, vary the configuration objects on and start the server takeover IP address on the new primary node if the entry in the cluster resource group indicates the configuration objects is to be varied on. If any of the devices in the cluster resource group are a primary auxiliary storage pool, all members of the auxiliary storage pool group will be varied on if the primary specifies the vary on value. The exit program is called on the new primary after the devices are moved to the new primary and varied on.
7. Cluster Resource Services submits batch job for each UDFS or primary auxiliary storage pool in the device list to vary the object on or off. The job is submitted to the job queue defined in the job description associated with the API's requesting user profile. The batch subsystem should be defined to allow these batch jobs to run concurrently in order to make switchover as fast as possible.
8. Set the cluster resource group status to Active (10) if the switchover to the new primary node is successful.
9. If the device entry in the cluster resource group indicates the device should be varied on and the vary on or the start of the server takeover IP address fails for some reason, the switchover will not complete successfully. The exit program will be called with an action code of Undo (15) and the devices will be moved back to the original primary node and/or the role of auxiliary storage pool will be swapped back as before (for geographic mirroring where the new primary node is at a different site than the current primary node). » After switching back all the devices, if all of the devices are varied-on successfully on the original primary node, clustering calls the exit program on the original primary node with an action code of Start.«
10. Set the cluster resource group status to Indoubt (30) if the devices cannot be successfully switched to the new primary node and cannot be returned to the same state on the old primary node.

When switching over cluster resource groups of different types, the order of switchover is important. Device cluster resource group objects should be done first followed by data cluster resource group objects and finally application cluster resource group objects.

If a cluster resource group has a status of Indoubt (30), the Start Cluster Resource Group API can be used to change the status to Active (10). See "Start Cluster Resource Group (QcstStartClusterResourceGroup) API" on page 164 API for more information.

This API requires:

1. Cluster Resource Services started on the node processing the request.
2. Cluster resource group status of Active (10).

This API operates in an asynchronous mode. See "Cluster Resource Services Job Structure" on page 220 for more information.

Restrictions:

- This API cannot be called from a cluster resource group exit program.
- » This API is not allowed for peer cluster resource groups.«

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

Cluster Resource Group Authority
*CHANGE

Cluster Resource Group Library Authority
*EXECUTE

Cluster Resource Group Lock
*EXCL

Exit Program Authority » (applies to user profile calling the API and user profile to run the exit program) «
*EXECUTE

Exit Program Library Authority » (applies to user profile calling the API and user profile to run the exit program) «
*EXECUTE

User Profile Authority » (applies to user profile to run the exit program) «
*USE

Request Information User Queue Authority
*OBJOPR, *ADD

Request Information User Queue Library Authority
*EXECUTE

Request Information User Queue Lock
*EXCLRD

Configuration Object Authority
*USE and *OBJMGT

Vary Configuration (VRYCFG) Command
*USE

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster containing the cluster resource group.

Cluster resource group name
INPUT; CHAR(10)

The name of the cluster resource group.

Exit program data
INPUT; CHAR(256)

256 bytes of data that is passed to the cluster resource group exit program when it is called. This parameter may contain any scalar data except pointers. For example, it can be used to provide state information. This data will be stored with the specified cluster resource group and copied to all nodes in the recovery domain. Pointers in this area will not resolve correctly on all nodes and should not be placed in the data. See "Cluster Resource Group Exit Program" on page 185 for information about the cluster resource group exit program. The data specified will replace the existing exit program data stored with the cluster resource group. If blanks are specified, then the exit program data stored with the cluster resource group will be cleared. This parameter must be set to *SAME if no exit program is specified for the cluster resource group. The following special value can be used:

**SAME* The exit program data stored with the cluster resource group specified will be passed to the exit program. This must be left justified.

Results information
INPUT; CHAR(30)

This parameter identifies a qualified user queue field and is followed by a reserved field.

Qualified user queue: Completion information is returned to this user queue, which exists on the node from which the API was called, after the function has completed. See the "Usage Notes" section of this API for a description of the data that is placed on this queue. This is a 20-character field. The first 10 characters contain the user queue name, and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, *CURLIB are not valid library names. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of the 30-character results information are reserved. Each character in this field must be set to hexadecimal zero.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See "Cluster APIs Use of User Queues" on page 221 and "Using Results Information" on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF18BA D	Error occurred with subsystem.
CPF2204 D	User profile &1 not found.
CPF26B6	Initialization program has ended with a hard error.
CPF2640	Vary command not processed.
CPF2659	Vary command may not have completed.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPF9801 D	Object &2 in library &3 not found
CPF9802 D	Not authorized to object &2 in &3.
CPF9803 D	Cannot allocate object &2 in library &3.
CPF9804 D	Object &2 in library &3 damaged.
CPF9810 D	Library &1 not found
CPFBB09 D	Cluster node &1 does not exist in cluster &2.
CPFBB0A D	Cluster node &1 in cluster &2 not active.
CPFBB0F D	Cluster resource group &1 does not exist in cluster &2.
CPFBB17 D	&1 API cannot be processed in cluster &2.
CPFBB18 D	Request &1 not allowed for cluster resource group &2.
CPFBB1E D	A switchover cannot be done for cluster resource group &1.
CPFBB2C D	Attributes for exit program &1 in library &2 are not valid.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB2E D	Job submission failed for cluster resource group &1 in cluster &2.
CPFBB38 D	Value &1 not allowed for library name.
CPFBB39 D	Current user does not have IOSYSCFG special authority.
CPFBB46 D	Cluster resource service internal error.
CPFBB5B D	Resource name &1 incorrect for configuration object &2 on node &3.
CPFBB66 D	Request failed for device cluster resource group &3.

Message ID	Message Text
CPFBB67 D	Ownership of hardware associated with configuration object &1 cannot be changed.
CPFBB69 A	Primary node &1 not current owner of hardware resource &2.
CPFBB6A D	Primary node &1 not current owner of specified devices.
CPFBB6C D	Hardware configuration is not complete for configuration objects in cluster resource group &1.
CPFBB6E E	Exit program data cannot be specified.
CPFBB7B D	Device type not correct for configuration object &1 on node &2.
CPFBB80 D	Request failed for device cluster resource group &3.
CPFBB90 D	Request failed for device cluster resource group &3.
CPFBB92 D	Hardware resource &1 not owned by node &3 or node &4.
CPFBB98 D	Hardware resource &1 not switchable.
CPFBB99 D	Request failed for device cluster resource group &3.
CPIBB10 D	Cluster resource group exit program &1 in library &2 on node &3 failed.
TCP1B01 D	Unable to determine if &1 interface started.
TCP1B02 D	Cannot determine if &1 interface started.
TCP1B05 D	&2 interface not started. Reason &1.
TCP1B10 D	&2 interface not started.
TCP1B11 D	&1 interface not started. Tried to exceed maximum number of active interfaces allowed.
TCP1B12 D	&1 interface not started. &1 interface already active.
TCP1B13 D	&1 interface not started. &1 interface not defined the TCP/IP configuration.
TCP1B14 D	&1 interface not started. Line description &2 not found.
TCP1B15 D	Line description &2 unusable. Internal errors encountered.
TCP1B16 D	&2 interface not started.
TCP1B25 D	&1 interface not started.
TCP265F D	INTNETADR parameter value &2 not valid.
TCP1B61 D	Unable to determine if &1 interface ended.&2 successful (&3).
TCP1B62 D	Cannot determine if &1 interface ended.
TCP1B65 D	&2 interface not ended. Reason &1.
TCP1B72 D	&1 interface not ended. &1 interface is not active.
TCP1B73 D	&1 interface not ended. &1 interface not defined in TCP/IP configuration.
TCP1B74 D	&1 interface not ended. Line description &2 not found.
TCP1B85 D	&1 interface not ended.
TCP3210 D	Connection verification statistics: &1 of &2 successful (&3).
TCP9999 D	Internal system error in program &1.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the "Usage Notes" on page 139 above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C1E E	Required parameter &1 omitted.
CPF3C39 E	Value for reserved field not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found
CPF9820 E	Not authorized to use library &1.
CPFBB02 E	Cluster &1 does not exist.

Message ID	Error Message Text
CPFBB09 E	Cluster node &1 does not exist in cluster &2.
CPFBB0A E	Cluster node &1 in cluster &2 not active.
CPFBB0F E	Cluster resource group &1 does not exist in cluster &2.
CPFBB1E E	A switchover cannot be done for cluster resource group &1.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB2C E	Attributes for exit program &1 in library &2 are not valid.
CPFBB32 E	Attributes of user queue &1 in library &2 are not valid.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB44 E	&1 API cannot be called from a cluster resource group exit program.
CPFBB6E E	Exit program data cannot be specified.
CPFBB3B E	Request not allowed for cluster resource group type &1.

API introduced: V4R4

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

List Cluster Resource Group Information (QcstListClusterResourceGroupIn) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Qualified user space name	Input	Char(20)
3	Format name	Input	Char(8)
4	Cluster name	Input	Char(10)
5	Cluster resource group name	Input	Char(10)
6	Results information	Input	Char(30)
7	Error code	I/O	Char(*)

Service Program: QCSTCRG3
Default Public Authority: *USE
Threadsafe: Yes

The List Cluster Resource Group Information (QcstListClusterResourceGroupIn) API returns the contents of a cluster resource group object.

This API can be called from a cluster resource group exit program. However if the status of the cluster resource group specified on the list request is Initialize Pending (540); this API will fail. The cluster resource group exit program will not be called when this API is run.

If Cluster Resource Services has not been started:

- The information returned may not be current.
- Information will only be returned for a cluster resource group on the node running the API.

If Cluster Resource Services has been started, this API will return information about the cluster resource group even if it does not exist on the node from which the API is called. For information retrieved from another node, the API will always indicate success. To determine if data was actually returned, wait for the API completion message which is sent to the results information queue and then check the length returned in the generic header portion of the user space.

If format LRGI0200 information about a device cluster resource group is requested and no configuration object entries have yet been added to the cluster resource group, the list entry size will be 0.

Authorities and Locks

Cluster Resource Group Authority
*USE

Cluster Resource Group Library Authority
*EXECUTE

Cluster Resource Group Lock
*SHRRD

User Space Authority
*CHANGE

User Space Library Authority
*EXECUTE

User Space Lock
*EXCLRD

Request Information User Queue Authority
*OBJOPR, *ADD

Request Information User Queue Library Authority
*EXECUTE

Request Information User Queue Lock
*EXCLRD

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the Results information parameter. If the cluster resource group exists on this node, then the Request handle field will be blank.

Qualified user space object
INPUT; CHAR(20)

The user space that is to receive the created list. The first 10 characters contain the user space name, and the second 10 characters contain the name of the library where the user space is located. No special values can used for the library name, for example, QTEMP, *CURLIB, or *LIBL.

The user space cannot be in an independent auxiliary storage pool.

Format name
INPUT; CHAR(8)

The format of the information to be returned. The format names supported are:

"LRGI0100 Recovery Domain Entries Section" on page 145	Recovery Domain Entries
"LRGI0200 Configuration Object Entries Section" on page 145	Configuration Object Entries

"LRGI0200 Configuration Object Entries Section" on page 145	Recovery Domain Entries with additional information on site name and data port IP addresses.

Cluster name
INPUT; CHAR(10)

The name of the cluster containing the cluster resource group.

Cluster resource group name
INPUT; CHAR(10)

The name of the cluster resource group for which information is retrieved.

Results information
INPUT; CHAR(30)

This parameter identifies a qualified user queue field and is followed by a reserved field. This parameter is ignored if the cluster resource group exists on the node initiating the API request.

Qualified user queue: Completion information is returned to this user queue, which exists on the node from which the API was called, after the function has completed. See the "Usage Notes" on page 151 section of this API for a description of the data that is placed on this queue. This is a 20-character field. The first 10 characters contain the user queue name, and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, *CURLIB are not valid library names. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of the 30-character results information are reserved. Each character in this field must be set to hexadecimal zero.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Format of the Generated Lists

The cluster resource group list consists of:

- A user space
- A generic header
- An input parameter section
- A header section
- A list data section:
  - LRGI0100 format,
  - LRGI0200 format
  - LRGI0300 format

For details about the user area and generic header, see User Space Format for List APIs. For details about the remaining items, see the following sections. For detailed descriptions of the fields in the list returned, see "Field Descriptions" on page 146.

When you retrieve list entry information from a user space, you must use the entry size returned in the generic header for the LRGI0100 or LRGI0200 formats. The size of each entry may be padded at the end. If you do not use the entry size, the result may not be valid. For examples of how to process lists, see Examples.

Input Parameter Section

A copy of most parameters coded in the call to the API.

Offset		Type	Field
Dec	**Hex**		
0	0	CHAR(10)	User space name
10	A	CHAR(10)	User space library name
20	14	CHAR(8)	Format name
28	1C	CHAR(10)	Cluster name
38	26	CHAR(10)	Cluster resource group name
48	30	CHAR(16)	Request handle
64	40	CHAR(30)	Results information
Note: The **Results information** field is comprised of the following 3 fields.			
64	40	CHAR(10)	Results information user queue name
74	4A	CHAR(10)	Results information user queue library name
84	54	CHAR(10)	Reserved

Header Section

Global information about the cluster resource group.

Offset		Type	Field
Dec	**Hex**		
0	0	CHAR(1)	Information status
1	1	CHAR(3)	Reserved
4	4	BINARY(4)	Cluster resource group type
8	8	BINARY(4)	Current cluster resource group status
12	C	CHAR(10)	Cluster resource group exit program name
22	16	CHAR(10)	Cluster resource group exit program library name
32	20	CHAR(8)	Cluster resource group exit program format name
40	28	CHAR(10)	User profile
50	32	CHAR(256)	Exit program data
306	132	CHAR(16)	Takeover IP address
322	142	CHAR(10)	Job name
332	14C	BINARY(4)	Allow application restart
336	150	BINARY(4)	Number of restarts
340	154	BINARY(4)	Previous cluster resource group status
344	158	CHAR(8)	Reporting node
352	160	CHAR(50)	Text description
402	192	CHAR(10)	Distribute information user queue name
412	19C	CHAR(10)	Distribute information user queue library name
422	1A6	CHAR(1)	Configure takeover IP address
423	1A7	CHAR(1)	Clustered hash table cluster resource group

Offset		Type	Field
Dec	Hex		
424	1A8	BINARY(4)	Failover wait time
428	1AC	BINARY(4)	Failover default action
432	1B0	CHAR(10)	Failover message queue name
442	1BA	CHAR(10)	Failover message queue library name
452	1C4	CHAR(1)	» Reserved «
» 453	1C5	CHAR(20)	Application id «
» 473	1D9	CHAR(1)	Allow active takeover IP address «

LRGI0100 Recovery Domain Entries Section

The cluster resource group recovery domain information is returned when LRGI0100 is specified as the format type. Each recovery domain entry returned will contain both the current and the preferred node role. For detailed descriptions of the fields in the table, see "Field Descriptions" on page 146.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(8)	Node id
8	8	BINARY(4)	Current node role
12	C	BINARY(4)	Membership status
16	10	BINARY(4)	Preferred node role

LRGI0200 Configuration Object Entries Section

The cluster resource group resilient device information is returned when LRGI0200 is specified as the format type. For detailed descriptions of the fields in the table, see "Field Descriptions" on page 146.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Configuration object name
10	A	CHAR(2)	Reserved
12	C	BINARY(4)	Configuration object type
16	10	BINARY(4)	Device type
20	14	BINARY(4)	Configuration object online
24	18	BINARY(4)	Device subtype
28	1C	CHAR(16)	Server takeover IP address

LRGI0300 Recovery Domain Entries Section

The cluster resource group recovery domain information is returned when LRGI0300 is specified as the format type. Each recovery domain entry returned will contain both the current and the preferred node role. For detailed descriptions of the fields in the table, see "Field Descriptions" on page 146.

Offset		Type	Field
Dec	Hex		
0	0	BIN(4)	Length of recovery domain entry
4	4	CHAR(8)	Node id
12	C	BINARY(4)	Current node role
16	10	BINARY(4)	Membership status
20	14	BINARY(4)	Preferred node role
24	18	CHAR(8)	Site name
32	20	BINARY(4)	Offset to data port IP address array
36	24	BINARY(4)	Number of data port IP addresses
*	*	Array(*) of CHAR(16)	Data Port IP address

Field Descriptions

Allow application restart. Attempt to restart an application if the cluster resource group exit program fails. Possible values are:

0 Do not attempt to restart the application. The cluster resource group exit program is called with an action code of Failover (9).

1 Attempt to restart the application on the same node. The cluster resource group exit program will be called with an action code of Restart (3). If the application cannot be restarted in the specified maximum number of attempts, the cluster resource group exit program will be called with an action code of Failover (9).

»

Allow active takeover IP address. Allows a takeover IP address to already be active when it is assigned to an application cluster resource group. This field is only valid when configure takeover IP address field is 0x01. Possible values are:

0 The takeover IP address must not already be active when starting the cluster resource group. This is the default value if the field is not specified.

1 The takeover IP address is allowed to be active prior to starting the cluster resource group but only on the primary node.

«

» Application id. This is a string to identify the application supplying the peer cluster resource group. The recommended format is 'vendor-id.name' where vendor-id is an identifier for the vendor creating the cluster resource group and name is the application name. For example, QIBM.ExamplePeer, indicates it is supplied by IBM for the ExamplePeer application. It is not recommended to use QIBM for vendor id name unless the cluster resource group is supplied by IBM. «

Clustered hash table (CHT) cluster resource group. Indicates if this cluster resource group is associated with the clustered hash table server.

0 Cluster resource group not associated with CHT server.

1 Cluster resource group associated with CHT server.

Cluster name. The name of the cluster containing the cluster resource group.

Cluster resource group exit program format name. Indicates which format should be used for the Information Given To User parameter on the cluster resource group exit program when it is called. See

"Cluster Resource Group Exit Program" on page 185 for details. This field will contain hexadecimal zeroes if no exit program is specified for the cluster resource group.

Cluster resource group exit program library name. The name of the library that contains the exit program for the cluster resource group. This field will contain hexadecimal zeroes if no exit program is specified for the cluster resource group.

Cluster resource group exit program name. The name of the user provided exit program. This field will contain *NONE left justified and padded with blanks if no exit program is specified for the cluster resource group.

Cluster resource group name. The name of the cluster resource group.

Cluster resource group type. The type of resilient resource information contained within the cluster resource group object.

1	Data resiliency
2	Application resiliency
3	Device resiliency
4	Peer resiliency

Configuration object name. The configuration object name for a device entry. For example, the auxiliary storage pool device description name.

Configuration object online. Vary the configuration object on and start the server takeover IP address or leave the configuration object varied off and the server takeover IP address inactive when a device is switched from one node to another with the "Initiate Switchover (QcstInitiateSwitchOver) API" on page 135 or when it is failed over to a backup node. This attribute does **not** vary the device on or off and does **not** start or end the server takeover IP address when the cluster resource group is started or ended and when adding a new device entry to the cluster resource group. For secondary auxiliary storage pools, only a value of 2 is valid. If cluster resources cannot determine if this value is correct for a device entry because the auxiliary storage pool is not yet created, any errors will be detected when the cluster resource group is started. A value of 2 cannot be specified for any other device type. Possible values are:

0	Do not vary the configuration object on and do not start the server takeover IP address.
1	Vary the configuration object on and start the server takeover IP address.
2	Perform the same action for a secondary auxiliary storage pool as is specified for the primary.

Configuration object type. This specifies the type of configuration object specified with configuration object name. Possible values are:

1	Device description

Configure takeover IP address. This field identifies who is responsible for configuring (adding and removing) the takeover IP address. This does not affect the starting and ending of the takeover IP address, Cluster Resource Services will perform this function. The following values are valid:

0x00	Cluster Resource Services is responsible for configuring the takeover IP address.
0x01	User is responsible for configuring the takeover IP address.

Current cluster resource group status. The current status of the cluster resource group when this information was gathered. See "Summary of Cluster Resource Group Status" on page 67 for the possible values and definitions of the status.

Current node role. The curent role the node has in the current recovery domain. For primary-backup model cluster resource groups a node can have one of three roles: primary, backup, or replicate. Only one node can be designated as the primary. Backup nodes are assigned a backup order. One indicates the first backup, two the second backup, and so on. Replicates are not ordered and cannot become a primary or backup node unless the "Change Cluster Resource Group (QcstChangeClusterResourceGroup) API" on page 86 is used to change its role from replicate to either a backup or primary.
For peer model cluster resource groups a node can have one of two roles: peer or replicate. Any number of nodes can be designated as the peer or replicate. Peer nodes are not ordered and can be an active access point for the cluster resources. Replicates are not ordered and cannot become an active access point for the cluster resource unless the "Change Cluster Resource Group (QcstChangeClusterResourceGroup) API" on page 86 is used to change its role from replicate to peer.

The following summarizes the valid values for this field:

0	Primary node. Only one node can have this value.
>=1	Backup node. The backup order is designated by increasing value.
-1	Replicate node. All replicates have this value.
	Peer node. All peers have this value.

-4

Data port IP address. The IP address associated with the recovery domain node.

Device subtype. The subtype of the device. This information is only as current as the last time the cluster resource group object could be updated. If configuration changes have been made on the node which owns the hardware and those changes have not yet been distributed to all nodes in the cluster, this information may be inaccurate. The data cannot be distributed if the configuration was changed on a node which does not have cluster resource services running.

-1	The subtype cannot be determined because hardware configuration is not complete.
0	This device type does not have a subtype.
1	UDFS independent auxiliary storage pool.
2	Secondary independent auxiliary storage pool.
3	Primary independent auxiliary storage pool.

Device type. The type of device description.

1	Auxiliary storage pool

Distribute information user queue library name. The name of the library that contains the user queue to receive distributed information. This field will contain hexadecimal zeroes if no user queue is specified for the cluster resource group.

Distribute information user queue name. The name of the user queue to receive distributed information from the "Distribute Information (QcstDistributeInformation) API" on page 127 API. This field will contain *NONE left justified and padded with blanks if no user queue is specified for the cluster resource group.

Exit program data. 256 bytes of data which is passed to the cluster resource group exit program when it is called. This field will contain hexadecimal zeroes if no exit program is specified for the cluster resource group.

Failover default action. Should a response to the failover message queue not be received in the failover wait time limit, then this field tells clustering what it should do pertaining to the failover request.

0	Proceed with failover.
1	Do NOT attempt failover.

Failover message queue library name. The name of the library that contains the user queue to receive failover messages. This field will contain hexadecimal zeroes if if no message queue is specified for the cluster resource group.

Failover message queue name. The name of the message queue to receive messages dealing with failover. This field will contain *NONE left justified and padded with blanks if no message queue is specified for the cluster resource group.

Failover wait time. Number of minutes to wait for a reply to the failover message that was enqueued on the failover message queue. Possible values are:

-1	Wait forever until a response is given to the failover inquiry message.
0	Failover proceeds without user intervention. Acts the same as V5R1M0 and prior.
>=1	Number of minutes to wait for a response to the failover inquiry message. If no response is received in the specified number of minutes, the failover default action field will be looked at to decide how to proceed.

Format name. The content and format of the information returned in the user space.

Job name. The name given the batch job that is submitted to call the cluster resource group exit program. This field will contain hexadecimal zeroes if no exit program is specified for the cluster resource group. Valid special values for this field are:

**JOBD*	The job name is determined from the job description specified in the user profile for the cluster resource group exit program.

Information status. Indicates the consistency of the retrieved information.

0	The information is consistent for all active nodes in the cluster.
1	The information retrieved from the node running the API may not be consistent with all active nodes in the cluster. In order to obtain consistent information: • Call this API on an active node in the cluster, if the node running the API is not active. • Start Cluster Resource Services on the node running the API if it is not active. • Use the Change Cluster Recovery (CHGCLURCY) CL command, if the cluster resource group job if the cluster resource group is not running but Cluster Resource Services is active on the node.

Length of recovery domain entry. The length in bytes of the entry for LRGI0300 format in the list data section. Use this field to access the next entry in the list.

Membership status. The cluster resource group membership status for the node.

0	Active	The node is an active member of the cluster resource group membership.
1	Inactive	The node is not an active member of the cluster resource group membership. The Cluster Resource Services may not be active on the node or the cluster resource group job could be ended on the node.
2	Partition	The node is partitioned from the other members of the cluster resource group membership.

3	Ineligible	The node is an active member of the cluster resource group membership but not eligible to become the cluster resource group primary node.

Node id. A unique string of characters that identifies a node that is in the recovery domain of the cluster resource group.

Number of data port IP addresses. The number of data port IP addresses associated with the recovery domain node.

Number of restarts. Number of times the cluster resource group exit program can be restarted on the same node before a failover occurs. Maximum number of restarts is 3. This field is valid only for an application cluster resource group. It will always contain hexadecimal zero for a data or device cluster resource group.

Offset to data port IP address array. The byte offset from the beginning of the user space to the first data port IP address.

Preferred node role. The preferred role the node has in the recovery domain. For primary-backup model cluster resource groups a node can have one of three roles: primary, backup, or replicate. Only one node can be designated as the primary. Backup nodes are assigned a backup order. One indicates the first backup, two the second backup, and so on. Replicates are not ordered and cannot become a primary or backup node unless the "Change Cluster Resource Group (QcstChangeClusterResourceGroup) API" on page 86 is used to change its role from replicate to either a backup or primary.
For peer model cluster resource groups a node can have one of two roles: peer or replicate. Any number of nodes can be designated as the peer or replicate. Peer nodes are not ordered and can be an active access point for the cluster resources. Replicates are not ordered and cannot become an active access point for the cluster resource unless the "Change Cluster Resource Group (QcstChangeClusterResourceGroup) API" on page 86 is used to change its role from replicate to peer.
Valid values are:

0	Primary node. Only one node can have this value.
>=1	Backup node. The backup order is designated by increasing value.
-1	Replicate node. All replicates have this value.
-4	Peer node. All peers have this value.

Previous cluster resource group status. Indicates the status of this cluster resource group before the current request was executed.

Reporting node. The cluster node which returned the list information for the cluster resource group Valid special values are:

**LOCAL*	The information was retrieved from a cluster resource group on the local system and the local system did not have a cluster node associated with it.

Request handle. Identifies this API call. It is used to associate this call to a response on the user queue specified in the Results Information parameter.

Reserved. The field will contain hexadecimal zeroes.

Results information. This field identifies a qualified user queue name and is followed by a reserved field.

Results information user queue library name. This field identifies the library where the results information user queue resides.

Results information user queue name. This field identifies the name of the user queue to which messages are sent when this API operates asynchronously.

Server takeover IP address. This is a takeover IP address for servers associated with the relational database name in the device description for an auxiliary storage pool. The address will be represented in dotted decimal format and returned as a null-terminated string. If not specified, or for a secondary and UDFS auxiliary storage pool, this field will contain *NONE left justified and padded with blanks. If the current cluster version is 2, this field is set to hexadecimal zeroes.

Site name. The name of the site associated with the recovery domain node. If the current cluster version is 3, this field is set to hexadecimal zeroes.

Takeover IP address. The floating IP address that is to be associated with the application. This field is only meaningful for an application cluster resource group. It is set to hexadecimal zeroes for a data or device cluster resource group. This field will be represented in dotted decimal format and returned as a null-terminated string.

Text description. The text description associated with the cluster resource group object.

User profile. The name of the user profile under which the cluster resource group exit program is run. This field will contain hexadecimal zeroes if no exit program is specified for the cluster resource group.

User space library name. The name of the library containing the user space.

User space name. The name of the user space which will contain the information retrieved from the cluster resource group.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See “Cluster APIs Use of User Queues” on page 221 and “Using Results Information” on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF2204 D	User profile &1 not found.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPF9801 D	Object &2 in library &3 not found.
CPF9802 D	Not authorized to object &2 in &3.
CPF9803 D	Cannot allocate object &2 in library &3.
CPF9804 D	Object &2 in library &3 damaged..
CPF9810 D	Library &1 not found.
CPFBB24 D	Node &1 not participating in &2 API protocol.
CPFBB32 D	Attributes of user queue &1 in library &2 are not valid.
CPFBB38 D	Library name &1 not allowed for this request.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the “Usage Notes” above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPFBB02 E	Cluster &1 does not exist.
CPFBB0F E	Cluster resource group &1 does not exist in cluster &2.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB32 E	Attributes of user queue &1 in library &2 are not valid.
CPFBB38 E	Library name &1 not allowed for this request.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB40 E	The value &1 specified for the allow application restarts parameter is not valid
CPFBB43 E	Format name &1 not valid for cluster resource group type &2.
CPFBB46 E	Cluster Resource Services internal error.
CPFBB70 E	API request &1 not compatible with current cluster version.

API introduced: V4R4

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

List Cluster Resource Groups (QcstListClusterResourceGroups) API

Required Parameter Group:

1	Qualified user space name	Input	Char(20)
2	Format name	Input	Char(8)
3	Cluster name	Input	Char(10)
4	Error code	I/O	Char(*)

Service Program: QCSTCRG3
Default Public Authority: *USE
Threadsafe: Yes

The List Cluster Resource Groups (QcstListClusterResourceGroups) API generates a list of and descriptive information about the cluster resource groups in the cluster specified by the cluster name parameter. The generated list is placed in the specified user space and replaces any existing list. Information will be returned for all cluster resource groups in the cluster, even if they do not exist on the node running the API. All the cluster resource groups will be returned in the generated list regardless of the authority of the user calling the API. The List Objects (QUSLOBJ) API can be used to provide a list of cluster resource group objects on this node only. Since the request for information is not distributed to other nodes in the cluster, the information about a cluster resource group that is returned shows the values obtained from the node running the API. Several conditions (for example, Cluster Resource Services not active on the node running the API) may produce inconsistent information about a cluster resource group in the cluster.

This API can be called from a cluster resource group exit program.

Authorities and Locks

User Space Authority
*CHANGE

User Space Library Authority
*EXECUTE

User Space Lock
*EXCLRD

Required Parameter Group

Qualified user space name
INPUT; CHAR(20)

The name of the *USRSPC object that is to receive the generated list. The first 10 characters contain the user space object name, and the second 10 characters contain the name of the library where the user space is located. No special values are supported for library name, for example, QTEMP, *CURLIB, or *LIBL.

The user space cannot be in an independent auxiliary storage pool.

Format Name
INPUT; CHAR(8)

The format of the information returned for each cluster resource group object. Possible values are:

"CRGL0100 Format" on page 154 Cluster resource group object names

Cluster name
INPUT; CHAR(10)

The name of the cluster for which the list of cluster resource group objects is retrieved.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Format of the Generated Lists

The cluster resource group list consists of:

- A user space
- A generic header
- An input parameter section
- A header section
- A list data section:
  - CRGL0100 format

For details about the user area and generic header, see User Space Format for List APIs. For detailed descriptions of the fields in the list returned, see "Field Descriptions" on page 154.

The completion code in the generic header should be checked to determine if the API completed successfully. When you retrieve list entry information from a user space, you must use the entry size returned in the generic header. The size of each entry may be padded at the end. If you do not use the entry size, the result may not be valid.

Input Parameter Section

An exact copy of the parameters coded in the call to the API.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	User space name
10	A	CHAR(10)	User space library name
20	14	CHAR(8)	Format name
28	1C	CHAR(10)	Cluster name

Header Section

Global information about the cluster resource groups.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(1)	Information status

CRGL0100 Format

General information about the cluster resource groups in the cluster. Detailed information about a single cluster resource group can be obtained by using the "List Cluster Resource Group Information (QcstListClusterResourceGroupIn) API" on page 141 API.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Cluster resource group name
10	A	BINARY(2)	Cluster resource group type
12	C	BINARY(4)	Cluster resource group status
16	10	CHAR(8)	Primary node id
» 24	18	CHAR(20)	Application id «

Field Descriptions

» Application id. This is a string to identify the application supplying the peer cluster resource group. The recommended format is 'vendor-id.name' where vendor-id is an identifier for the vendor creating the cluster resource group and name is the application name. For example, QIBM.ExamplePeer, indicates it is supplied by IBM for the ExamplePeer application. It is not recommended to use QIBM for vendor id name unless the cluster resource group is supplied by IBM. «

Cluster name. The name of the cluster for which the list of cluster resource groups is to be retrieved.

Cluster resource group name. The name of the cluster resource group.

Cluster resource group status. The status of the cluster resource group when this information was gathered. See "Summary of Cluster Resource Group Status" on page 67 for the possible values and definitions of the status.

Cluster resource group type. The type of the cluster resource group. The possible values are:

1 Data resiliency.
2 Application resiliency.
3 Device resiliency.
4 Peer resiliency.

Format name. The content and format of the information returned for each cluster resource group object entry. The value must be set to CRGL0100.

Information status. Indicates the consistency of the retrieved information.

0 The information is consistent for all active nodes in the cluster.
1 The information retrieved from the node running the API may not be consistent with all active nodes in the cluster. In order to obtain consistent information:

- Call this API on an active node in the cluster, if the node running the API is not active.
- Start Cluster Resource Services on the node running the API if it is not active.

Primary node id. A unique string of characters that identifies the node that has a current node role of primary for the cluster resource group object.

*NONE Valid only for peer cluster resource group. There is no primary node for a peer cluster resource group.

User space library name. The name of the library that contains the user space.

User space name. The name of the user space that receives the list.

Error Messages

Messages that are delivered through the error code parameter are listed here.

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3C21 E	Format name &1 is not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF9810 E	Library &1 not found.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.

API introduced: V4R4

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Remove Cluster Resource Group Device Entry (QcstRmvClusterResourceGroupDev) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Cluster resource group name	Input	Char(10)
4	Configuration object entry information	Input	Char(*)
5	Format name	Input	Char(8)
6	Results information	Input	Char(30)
7	Error code	I/O	Char(*)

Service Program: QCSTCRG1
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Remove Cluster Resource Group Device Entry (QcstRmvClusterResourceGroupDev) API removes one or more configuration objects from a device cluster resource group. All configuration object entries can be removed but at least one configuration object entry must exist before the "Start Cluster Resource Group (QcstStartClusterResourceGroup) API" on page 164 API can be called.

Ownership of the hardware associated with the configuration object being removed is not affected. The hardware is still owned by whatever node owned it before this API was called.

If an exit program is specified for the cluster resource group, the cluster resource group exit program is called with an action code of Remove Device Entry (18) on all active nodes in the recovery domain. The cluster resource group status is set to Remove Device Entry Pending (600). If the exit program completes successfully, the cluster resource group status is reset to its value at the time the API was called. If the exit program fails and the cluster resource group cannot be restored to its original condition, the cluster resource group status is set to Indoubt (30).

Removing a device from a cluster resource group does not vary the device off.

This API requires:

1. Cluster Resource Services must be active on the node processing the request.
2. If an exit program is specified, the exit program must exist on all nodes in the recovery domain.
3. At least one node in the recovery domain must be active.
4. If the cluster resource group is active, all members of an auxiliary storage pool group must be removed at the same time.
5. If the cluster resource group is active, the last device entry cannot be removed. The cluster resource group must be ended first.

This API operates in an asynchronous mode. See "Behavior of Cluster Resource Services APIs" on page 221 for more information.

Restriction: This API cannot be called from a cluster resource group exit program.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

Cluster Resource Group Authority
*CHANGE

Cluster Resource Group Library Authority
*EXECUTE

Cluster Resource Group Lock
*EXCL

Exit Program Authority » (applies to user profile calling the API and user profile to run the exit program) «
*EXECUTE

Exit Program Library Authority » (applies to user profile calling the API and user profile to run the exit program) «
*EXECUTE

User Profile Authority » (applies to user profile to run the exit program) «
*USE

Request Information User Queue Authority
*OBJOPR, *ADD

Request Information User Queue Library Authority
*EXECUTE

Request Information User Queue Lock
*EXCLRD

» *Configuration object*
*USE and *OBJMGT «

» *Configuration Object Lock*
*EXCLRD «

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster to which the cluster resource group belongs.

Cluster resource group name
INPUT; CHAR(10)

The name of the cluster resource group which is to be changed.

Configuration object entry information
INPUT; CHAR(*)

Detailed information about the configuration objects to be added to the cluster resource group. For more information, see "Device Resiliency (RGDR0100 Format)" on page 158.

Format name
INPUT; CHAR(8)

The content and format of the configuration object information. The possible format names are:

"Device Resiliency (RGDR0100 Format)" on page 158	This format describes the resilient device.

Results information

INPUT; CHAR(30)

This parameter identifies a qualified user queue field and is followed by a reserved field.

Qualified user queue: Completion information is returned to this user queue, which exists on the node from which the API was called, after the function has completed. See the "Usage Notes" on page 159 section of this API for a description of the data that is placed on this queue. This is a 20-character field. The first 10 characters contain the user queue name, and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, *CURLIB are not valid library names. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of the 30-character results information are reserved. Each character in this field must be set to hexadecimal zero.

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Device Resiliency (RGDR0100 Format)

Offset		Type	Field
Dec	**Hex**		
0	0	BINARY(4)	Offset to configuration object array
4	4	BINARY(4)	Number of entries in configuration object array
8	8	BINARY(4)	Length of configuration object array entry
12	C	BINARY(4)	Offset to additional fields
16	10	BINARY(4)	Length of additional fields
*	*	Array (*) of CHAR(16)	Configuration object array
These fields repeat, in the order listed, for each device.		CHAR(10)	Configuration object name
		CHAR(2)	Reserved
		BINARY(4)	Configuration object type

Field Descriptions

Configuration object array. This array identifies the resilient devices.

Configuration object name. The name of the auxiliary storage pool device description object which is to be removed from the cluster resource group.

Configuration object type. This specifies the type of configuration object specified with configuration object name. Possible values are:

1 Device description

Length of additional fields. The length in bytes of additional fields. This must be set to hexadecimal zero. It will be used in a future release if more fields are needed in the RGDR0100 format.

Length of configuration object array entry. This specifies the length of an entry in the configuration object array.

Number of entries in configuration object array. The number of entries in the configuration object array. This must be greater than zero and less than or equal to 256.

Offset to additional fields. The byte offset from the beginning of this parameter to additional fields. This must be set to hexadecimal zero. It will be used in a future release if more fields are needed in the RGDR0100 format.

Offset to configuration object array. The byte offset from the beginning of this parameter to the configuration object array field.

Reserved. Must contain hexadecimal zeroes.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See "Cluster APIs Use of User Queues" on page 221 and "Using Results Information" on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Error Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF18BA D	Error occurred with subsystem.
CPF2113 E	Cannot allocate library &1.
CPF2204 D	User profile &1 not found.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPF9801 D	Object &2 in library &3 not found.
CPF9802 D	Not authorized to object &2 in &3.
CPF9803 D	Cannot allocate object &2 in library &3.
CPF9804 D	Object &2 in library &3 damaged.
CPF9810 D	Library &1 not found.
CPFBB0F D	Cluster resource group &1 does not exist in cluster &2.
CPFBB17 D	&1 API cannot be processed in cluster &2.
CPFBB18 D	Request &1 is not allowed for cluster resource group &2.
CPFBB2C D	Attributes of exit program &1 in library &2 are not valid.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB2E D	Job submission failed for cluster resource group &1 in cluster &2.
CPFBB32 D	Attributes of user queue &1 in library &2 are not valid.
CPFBB35 D	The user profile name &1 is not valid for this request.
CPFBB38 D	Library name &1 is not allowed for this request.
CPFBB39 D	The current user does not have IOSYSCFG special authority.
CPFBB46 D	Cluster Resource Services internal error.
CPFBB5B D	Resource name &1 incorrect for configuration object &2 on node &3.
CPFBB7B D	Device type not correct for configuration object &1 on node &2.
CPFBB98 D	Hardware resource &1 not switchable.
CPFBB9C D	Not all auxiliary storage pool group members added or removed together.
CPIBB10 D	Cluster Resource Group exit program &1 in library &2 on node &3 failed.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the "Usage Notes"

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF2204 E	User profile &1 not found.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C29 E	Object name &1 is not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB0F E	Cluster resource group &1 does not exist in cluster &2.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB2C E	Attributes of exit program &1 in library &2 are not valid.
CPFBB32 E	Attributes of user queue &1 in library &2 not valid.
CPFBB35 E	The user profile name &1 is not valid for this request.
CPFBB38 E	Library name &1 is not allowed for this request.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB43 E	Invalid format name &1 for cluster resource group type &2.
CPFBB44 E	&1 API cannot be called from a cluster resource group exit program.
CPFBB5F E	Number of configuration object entries not valid.
CPFBB60 E	Offset to configuration object array is not valid.
CPFBB61 E	Configuration object &1 specified more than once in configuration object array.
CPFBB63 E	The value specified for the field at offset &1 of configuration object array entry &2 is not valid.
CPFBB6B E	Request not valid for type &1 cluster resource group.
CPFBB6D E	Configuration object &1 not in cluster resource group &2.
CPFBB6F E	Last device entry cannot be removed from cluster resource group &1.
CPFBB70 E	API request &1 not compatible with current cluster version.

API introduced: V5R1

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Remove Node From Recovery Domain (QcstRemoveNodeFromRcvyDomain) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Cluster resource group name	Input	Char(10)
4	Node id	Input	Char(8)
5	Results information	Input	Char(30)
6	Error code	I/O	Char(*)

Service Program: QCSTCRG1
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Remove Node From Recovery Domain (QcstRemoveNodeFromRcvyDomain) API is used to remove a node from the recovery domain of a cluster resource group. The node being removed does not need to be active in the cluster to be removed from the recovery domain. For primary-backup model, if the cluster resource group has no backup nodes in either the current recovery domain or the preferred recovery domain, the primary node cannot be removed.

This API will do the following for all cluster resource group types:

1. Set the cluster resource group status to Remove Node Pending (550).
2. Call the exit program on all active nodes in the recovery domain with an action code of Remove Node (12), if an exit program is specified for the cluster resource group.
3. Reset the cluster resource group status to the value at the time the API was called, if the exit program completes successfully on all nodes.
4. Set the cluster resource group status to Indoubt (30) if the exit program fails on any node and the original state of the cluster resource group cannot be recovered.
5. Delete the cluster resource group object from the node removed.

This API will do the following for resilient application cluster resource groups:

1. If Cluster Resource Services configured the takeover IP address, it will be removed.

This API will do the following for resilient device cluster resource groups:

1. If the node being removed is the current primary node, ownership of the devices specified in the cluster resource group are switched from the current primary to the new primary, if none of the configuration objects are varied on on the current primary. For cross-site mirroring, the devices are switched to the new primary node (if the new primary node is at the same site as the current primary node) or to the active backup node at the same site as the current primary node. If any configuration objects are varied on, an error message is returned. In addition, the new primary node must be active.

   All members of an auxiliary storage pool group must be configured in the cluster resource group before ownership can be changed.

   Devices are not varied on after the ownership is switched. The node which is to become the new primary must be active in the cluster.

This API requires:

1. Cluster Resource Services started on the node running the API.
2. For primary-backup model cluster resource groups:
   - A cluster resource group status other than Active (10) in order to remove the node that is currently the primary.
   - At least one backup node in the recovery domain of the cluster resource group, if the primary node is removed.
3. For peer model cluster resource groups:
   - The last node designated as peer cannot be removed if the cluster resource group is Active (10).
4. At least one active node in the recovery domain of the cluster resource group after the successful completion of the remove operation.

This API operates in an asynchronous mode. See “Cluster Resource Services Job Structure” on page 220 for more information.

Restriction: This API cannot be called from a cluster resource group exit program.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

Cluster Resource Group Authority
*CHANGE, *OBJEXIST

Cluster Resource Group Library Authority
*EXECUTE

Cluster Resource Group Lock
*EXCL

Exit Program Authority » (applies to user profile calling the API and user profile to run the exit program) «
*EXECUTE

Exit Program Library Authority » (applies to user profile calling the API and user profile to run the exit program) «
*EXECUTE

User Profile Authority » (applies to user profile to run the exit program) «
*USE

Request Information User Queue Authority
*OBJOPR, *ADD

Request Information User Queue Library Authority
*EXECUTE

Request Information User Queue Lock
*EXCLRD

Configuration Object Authority
*USE and *OBJMGT

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster containing the cluster resource group.

Cluster resource group name
INPUT; CHAR(10)

The name of the cluster resource group from which the node will be removed.

Node id
INPUT; CHAR(8)

A unique string of characters that identifies the node that is to be removed from the recovery domain of the cluster resource group.

Results information
INPUT; CHAR(30)

This parameter identifies a qualified user queue field and is followed by a reserved field.

Qualified user queue: Completion information is returned to this user queue, which exists on the node from which the API was called, after the function has completed. See the “Usage Notes” on page 163 section of this API for a description of the data that is placed on this queue. This is a 20-character field. The first 10 characters contain the user queue name, and the second 10

characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, *CURLIB are not valid library names. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of the 30-character results information are reserved. Each character in this field must be set to hexadecimal zero.

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See "Cluster APIs Use of User Queues" on page 221 and "Using Results Information" on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF18BA D	Error occurred with subsystem.
CPF2113 E	Cannot allocate library &1.
CPF2204 D	User profile &1 not found.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPF9801 D	Object &2 in library &3 not found.
CPF9802 D	Not authorized to object &2 in &3.
CPF9803 D	Cannot allocate object &2 in library &3.
CPF9804 D	Object &2 in library &3 damaged.
CPF9810 D	Library &1 not found.
CPFBB09 D	Cluster node Id &1 does not exist in Cluster &2.
CPFBB0A D	Cluster node &1 not active in cluster &2.
CPFBB0B D	Request using takeover IP address &1 failed.
CPFBB0F D	Cluster resource group &1 does not exist in cluster &2.
CPFBB17 D	&1 API cannot be processed in cluster &2.
CPFBB18 D	Request &1 not allowed for cluster resource group &2.
CPFBB1B D	Cluster node &1 does not exist in the recovery domain for cluster resource group &2.
CPFBB2C D	Attributes of exit program &1 in library &2 are not valid.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB2E D	Job submission failed for cluster resource group &1 in cluster &2.
CPFBB32 D	Attributes of user queue &1 in library &2 are not valid.
CPFBB38 D	Library name &1 not allowed for this request.
CPFBB39 D	Current user does not have IOSYSCFG special authority.
CPFBB46 D	Cluster resource service internal error.
CPFBB50 D	Cluster node &1 not removed from cluster resource group &2.
CPFBB5B D	Resource name &1 incorrect for configuration object &2 on node &3.
CPFBB66 D	Request failed for device cluster resource group &3.
CPFBB67 D	Node &1 cannot take ownership of configuration object &2.
CPFBB69 D	Primary node &1 not current owner of hardware resource &2.
CPFBB6C D	Hardware configuration is not complete for configuration objects in cluster resource group &1.
CPFBB70 D	API request &1 not compatible with current cluster version.
CPFBB7B D	Device type not correct for configuration object &1 on node &2.
CPFBB80 D	Request failed for device cluster resource group &3.
CPFBB81 D	New primary node &1 not active.

Message ID	Message Text
CPFBB90 D	Request failed for device cluster resource group &3.
CPFBB92 D	Hardware resource &1 not owned by node &3 or node &4.
CPFBB98 D	Hardware resource &1 not switchable.
CPFBB99 D	Request failed for device cluster resource group &3.
CPFBB9B D	Auxiliary storage pool group member &1 not specified.
CPIBB10 D	Cluster resource group exit program &1 in library &2 on node &3 failed.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the "Usage Notes" on page 163 above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C1E E	Required parameter &1 omitted.
CPF3C39 E	Value for reserved field not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFBB02 E	Cluster &1 does not exist.
CPFBB09 E	Cluster node &1 does not exist in Cluster &2.
CPFBB0A E	Cluster node &1 not active in cluster &2.
CPFBB0F E	Cluster resource group &1 does not exist in cluster &2.
CPFBB1B E	Cluster node &1 does not exist in the recovery domain for cluster resource group &2.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB2C E	Attributes of exit program &1 in library &2 are not valid.
CPFBB32 E	Attributes of user queue &1 in library &2 are not valid.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB42 E	&1 API cannot be used within a cluster resource group exit program.
CPFBB50 E	Cluster node &1 not removed from cluster resource group &2.

API introduced: V4R4

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Start Cluster Resource Group (QcstStartClusterResourceGroup) API

Required Parameter Group:

1	Request handle	Output	Char(16)
2	Cluster name	Input	Char(10)
3	Cluster resource group name	Input	Char(10)
4	Exit program data	Input	Char(256)
5	Results information	Input	Char(30)
6	Error code	I/O	Char(*)

Service Program: QCSTCRG2
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Start Cluster Resource Group (QcstStartClusterResourceGroup) API will enable resiliency for the specified cluster resource group. The Start Cluster Resource Group API can be used to change cluster resource group from a status of Indoubt (30) or Inactive (20) to Active (10).

This API will do the following for all cluster resource group types:

1. Set the cluster resource group status to Start Cluster Resource Group Pending (560).
2. » For primary-backup model cluster resource groups, « if the current recovery domain has more than one backup node and some backup nodes are not active, the recovery domain may be reordered so that all active backup nodes are ordered before inactive backup nodes. If the inactive backup nodes are already ordered after active backup nodes or if there are no inactive backup nodes, Start will not change the recovery domain.
3. Call the exit program on all active nodes in the recovery domain with an action code of Start (2), if an exit program is specified for the cluster resource group.
4. Set the cluster resource group status to Active (10) if the exit program is successful on all active nodes in the recovery domain.
5. Perform the following on all nodes if the exit program is unsuccessful on any active node in the recovery domain:
   - Set the status of the cluster resource group to Indoubt (30).

This API will do the following for resilient application cluster resource groups:

1. Verify the takeover IP address has been configured on all nodes in the recovery domain except replicates.
2. Start the takeover IP address on the primary node.
3. If the exit program returns a failure, on the primary node:
   - Cancel the exit program job with an option of *IMMED.
   - End the takeover IP address on the primary node.

» This API will do the following for peer cluster resource groups:

1. All nodes that are designated with a peer role will be an active access point upon successful completion of this API. «

This API requires the following for all cluster resource group types:

1. Cluster Resource Services active on the node running the API.
2. » For primary-backup model cluster resource groups, the membership « status of the node currently assigned the role of primary must be active.
3. » For peer model cluster resource groups:
   - At least one node must be designated as peer or the cluster resource group cannot be started.
   - At least one peer node must have a membership status of Active. «
4. A cluster resource group status of Inactive (20) or Indoubt(30).

This API requires the following for resilient device cluster resource groups:

1. The cluster resource group must have at least one configuration object entry.
2. The configuration objects specified for the cluster resource group must exist on all active nodes in the recovery domain and the resource name specified in a configuration object must be the same on all active nodes in the recovery domain.
3. If a data base name has been specified for a configuration object, it must be the same on all active nodes in the recovery domain.

4. If a server takeover IP address is specified, it must exist on all nodes in the recovery domain.
5. The primary node must be the current owner of all IOPs or high-speed link I/O bridges for the devices in the cluster resource group.
6. Hardware configuration must be complete so that the physical hardware has been associated with the configuration object. In particular for auxiliary storage pools, the disk units must have been assigned.
7. The IOP or high-speed link I/O bridge controlling the devices specified in the cluster resource group must be accessible by all active nodes in the cluster resource group's recovery domain or by all nodes within the same site (for cross-site mirroring).
8. Starting the cluster resource group will not vary on the configuration objects or start the server takeover IP address.
9. All members of an auxiliary storage pool group must be configured in the cluster resource group.
10. A value of 2 for the device's 'configuration object online' attribute can be specified only for a secondary auxiliary storage pool.
11. If data port IP addresses are specified, it must exist on the node in the recovery domain.

» This API requires the following for application cluster resource groups: «

1. For an application cluster resource group, the exit program is not expected to complete on the primary node. The status of the cluster resource group will be set to Active (10) when the exit program job has been started on the primary and the exit program has completed successfully on all other nodes in the recovery domain.

This API operates in an asynchronous mode. See "Cluster Resource Services Job Structure" on page 220 for more information.

Restriction: This API cannot be called from a cluster resource group exit program.

Authorities and Locks

The program that calls this API must be running under a user profile with *IOSYSCFG special authority.

Cluster Resource Group Authority
*CHANGE

Cluster Resource Group Library Authority
*EXECUTE

Cluster Resource Group Lock
*EXCL

Exit Program Authority » **(applies to user profile calling the API and user profile to run the exit program)** «
*EXECUTE

Exit Program Library Authority » **(applies to user profile calling the API and user profile to run the exit program)** «
*EXECUTE

User Profile Authority » **(applies to user profile to run the exit program)** «
*USE

Request Information User Queue Authority
*OBJOPR, *ADD

Request Information User Queue Library Authority
*EXECUTE

Request Information User Queue Lock
*EXCLRD

Configuration Object Authority
*USE and *OBJMGT

Configuration Object Lock
*EXCLRD

Required Parameter Group

Request handle
OUTPUT; CHAR(16)

A unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter.

Cluster name
INPUT; CHAR(10)

The name of the cluster containing the cluster resource group.

Cluster resource group name
INPUT; CHAR(10)

The name of the cluster resource group which will be started.

Exit program data
INPUT; CHAR(256)

256 bytes of data that is passed to the cluster resource group exit program when it is called. This parameter may contain any scalar data except pointers. For example, it can be used to provide state information. This data will be stored with the specified cluster resource group and copied to all nodes in the recovery domain. Pointers in this area will not resolve correctly on all nodes and should not be placed in the data. See "Cluster Resource Group Exit Program" on page 185 for information about the cluster resource group exit program. The data specified will replace the existing exit program data stored with the cluster resource group before the exit program is called. If blanks are specified, then the exit program data stored with the cluster resource group will be cleared. This parameter must be set to *SAME if no exit program is specified. The following special value can be used:

**SAME* The exit program data stored with the cluster resource group specified will be passed to the exit program. This must be left justified.

Results information
INPUT; CHAR(30)

This parameter identifies a qualified user queue field and is followed by a reserved field.

Qualified user queue: Completion information is returned to this user queue, which exists on the node from which the API was called, after the function has completed. See the "Usage Notes" on page 168 section of this API for a description of the data that is placed on this queue. This is a 20-character field. The first 10 characters contain the user queue name, and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, *CURLIB are not valid library names. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of the 30-character results information are reserved. Each character in this field must be set to hexadecimal zero.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Usage Notes

Results Information User Queue

Asynchronous results are returned to a user queue specified by the Results Information parameter of the API. See “Cluster APIs Use of User Queues” on page 221 and “Using Results Information” on page 223 for details on how to create the results information user queue, the format of the entries, and how to use the data placed on the queue. The data is sent to the user queue in the form of a message identifier and the substitution data for the message (if any exists). The following identifies the data sent to the user queue (excluding the message text).

Message ID	Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF18BA D	Error occurred with subsystem.
CPF2113 E	Cannot allocate library &1.
CPF2204 D	User profile &1 not found.
CPF9801 D	Object &2 in library &3 not found.
CPF9802 D	Not authorized to object &2 in &3.
CPF9803 D	Cannot allocate object &2 in library &3.
CPF9804 D	Object &2 in library &3 damaged.
CPF9810 D	Library &1 not found.
CPFBB0A D	Node &1 is not active in cluster &2.
CPFBB0B D	Request using takeover IP address &1 failed.
CPFBB0F D	Cluster resource group &1 does not exist in cluster &2.
CPFBB17 D	&1 API cannot be processed in cluster &2.
CPFBB18 D	Request &1 not allowed for cluster resource group &2.
CPFBB2C D	Attributes of exit program &1 in library &2 are not valid.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB2E D	Job submission failed for cluster resource group &1 in cluster &2.
CPFBB32 D	Attributes of user queue &1 in library &2 are not valid.
CPFBB47 D	Cluster Resource Services ended abnormally.
CPFBB48 D	Cluster Resource Services error detected.
CPFBB5B D	Resource name &1 incorrect for configuration object &2 on node &3.
CPFBB66 D	Request failed for device cluster resource group &3.
CPFBB67 D	Ownership of hardware associated with configuration object &1 cannot be changed.
CPFBB68 D	Cluster resource group &1 has no configuration object entries.
CPFBB69 D	Primary node &1 not current owner of configuration object &2.
CPFBB6C D	Hardware configuration is not complete.
CPFBB6E D	Exit program data cannot be specified.
CPFBB70 D	API request &1 not compatible with current cluster version.
CPFBB7B D	Device type incorrect for configuration object &1 on node &2.
CPFBB80 D	Request failed for device cluster resource group &3.
CPFBB90 D	Request failed for device cluster resource group &3.
CPFBB92 D	Hardware resource &1 not owned by node &3 or node &4.
CPFBB98 D	Hardware resource &1 not switchable.
CPFBB99 D	Request failed for device cluster resource group &3.
CPFBB9A D	Online value not valid for device &1.
CPFBB9B D	Auxiliary storage pool group member &1 not specified.
CPFBB9E D	Data base name &1 not correct for configuration object &2 on node &3.
CPFBBA6 E	Server takeover IP address cannot be associated with device subtype &1.
CPIBB10 D	Cluster resource group exit program &1 in library &2 on node &3 failed.
TCP1B01 D	Unable to determine if &1 interface started.
TCP1B02 D	Cannot determine if &1 interface started.
TCP1B05 D	&2 interface not started. Reason &1.
TCP1B10 D	&2 interface not started.
TCP1B11 D	&1 interface not started. Tried to exceed maximum number of active interfaces allowed.

Message ID	Message Text
TCP1B12 D	&1 interface not started. &1 interface already active.
TCP1B13 D	&1 interface not started. &1 interface not defined the takeover IP configuration.
TCP1B14 D	&1 interface not started. Line description &2 not found.
TCP1B15 D	Line description &2 unusable. Internal errors encountered.
TCP1B16 D	&2 interface not started.
TCP1B25 D	&1 interface not started.
TCP9999 D	Internal system error in program &1.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the "Usage Notes" on page 168 above.

Message ID	Error Message Text
CPF2113 E	Cannot allocate library &1.
CPF2204 E	User profile &1 not found.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C1E E	Required parameter &1 omitted.
CPF3C39 E	Value for reserved field not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF980C E	Object &1 in library &2 cannot be in an independent auxiliary storage pool.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPFBB02 E	Cluster &1 does not exist.
CPFBB0A E	Node &1 is not active in cluster &2.
CPFBB0F E	Cluster resource group &1 does not exist in cluster &2.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB2C E	Attributes of exit program &1 in library &2 are not valid.
CPFBB32 D	Attributes of user queue &1 in library &2 are not valid.
CPFBB38 E	Library name &1 not allowed for this operation.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB44 E	&1 API cannot be called from a cluster resource group exit program.
CPFBB46 E	Cluster Resource Services internal error.
CPFBB6E E	Exit program data cannot be specified.

API introduced: V4R4

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Clustered Hash Table APIs

The information provided here includes:

Clustered Hash Table APIs—Introduction

The clustered hash table APIs enable sharing and replicating of data between cluster nodes. The data is stored in nonpersistent 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.

The clustered hash table server is started using the STRCHTSVR command. A cluster must be active on the node performing the start function. Using this command, the user defines the domain of the clustered hash table. A cluster resource group (CRG) (same name as clustered hash table server) is created to manage the domain of the clustered hash table.

There are two levels of security supported by a clustered hash table. One security level is associated with a clustered hash table server. This security is provided through the authorization list parameter on the STRCHTSVR command. This provides the ability to specify users that are allowed to start, end and connect to a clustered hash table server. For more details on the authorization list see the AUTL parameter on the STRCHTSVR command.

The second security level is provided on an entry stored in a clustered hash table. The authority access level is specified when an entry is stored in a clustered hash table. This provides the ability to restrict access to retrieving and updating an entry. For more details on the authority access level for an entry see the “Store Clustered Hash Table Entry (QcstStoreCHTEntry) API” on page 182 API.

First a client must connect to the clustered hash table server using “Connect Clustered Hash Table (QcstConnectCHT) API” on page 171 API. This establishes the communication infrastructure so clients can send requests to the clustered hash table server. A connection handle is returned to the client. This connection handle is required to be used on subsequent requests to the clustered hash table server. The client needs to connect from each job that is intending on communicating to the clustered hash table server.

Clients can disconnect from the clustered hash table server using the “Disconnect Clustered Hash Table (QcstDisconnectCHT) API” on page 172 API. The disconnect is necessary to cleanup the infrastructure established by the connection request. The disconnect is a local only request. It is recommended to use disconnect when done with the connection.

An entry is associated with a key and stored in a clustered hash table using the “Store Clustered Hash Table Entry (QcstStoreCHTEntry) API” on page 182 API. The key can be generated using the “Generate Clustered Hash Table Key (QcstGenerateCHTKey) API” on page 173 API or the user can generate their own.

The storage for the clustered hash table is not persistent. Not persistent means the storage for the clustered hash table is only known to the server on the local node and only available until the clustered hash table server is ended. All requests to store entries are replicated to other nodes in the clustered hash table domain. When an entry is stored, a time to live value is specified. The entry can become expired, when the time to live value has expired. Expired entries will be removed when processing various functions. For example, when adding another cluster node to the domain of an existing clustered hash table server. The existing clustered hash table entries, if any, are replicated to the cluster hash table domain node being added. Expired entries are removed from the clustered hash table during this process.

An entry is retrieved from the hash table by key using the “Retrieve Clustered Hash Table Entry (QcstRetrieveCHTEntry) API” on page 179 API. The retrieve request is only processed on the clustered hash table domain node requesting the function.

A list of all keys stored in a clustered hash table server can be obtained by using the “List Clustered Hash Table Keys (QcstListCHTKeys) API” on page 174 API. The list request is only processed on the clustered hash table domain node requesting the function.

The clustered hash table server is ended using the ENDCHTSVR CL command. The clustered hash table server will be ended on the cluster nodes specified.

The clustered hash table APIs have associated java classes. See ClusteredHashTable class for details.

Clustered Hash Table API List

The clustered hash table APIs are:

- "Connect Clustered Hash Table (QcstConnectCHT) API" (QcstConnectCHT) establishes a connection to the clustered hash table server specified.
- "Disconnect Clustered Hash Table (QcstDisconnectCHT) API" on page 172 (QcstDisconnectCHT) disconnects the using job from the clustered hash table server.
- "Generate Clustered Hash Table Key (QcstGenerateCHTKey) API" on page 173 (QcstGenerateCHTKey) returns a universally unique key that can be used to store an entry into the clustered hash table.
- "List Clustered Hash Table Keys (QcstListCHTKeys) API" on page 174 (QcstListCHTKeys) generates a list of keys and descriptive information about the entries stored in the clustered hash table specified by the connection handle parameter.
- "Retrieve Clustered Hash Table Entry (QcstRetrieveCHTEntry) API" on page 179 (QcstRetrieveCHTEntry) retrieves an entry from the clustered hash table specified by the connection handle parameter. The entry to be retrieved is identified by the key parameter.
- "Store Clustered Hash Table Entry (QcstStoreCHTEntry) API" on page 182 (QcstStoreCHTEntry) stores an entry in the clustered hash table identified by the connection handle.

Top | "Cluster APIs," on page 1 | APIs by category

Connect Clustered Hash Table (QcstConnectCHT) API

Required Parameter Group:

1	Connection handle	Output	Char(16)
2	Clustered hash table server name	Input	Char(10)
3	Error code	I/O	Void(*)

Service Program: QCSTCHT
Default Public Authority: *USE
Threadsafe: Yes

The Connect Clustered Hash Table (QcstConnectCHT) API establishes a connection to the clustered hash table server specified. This API returns a connection handle that will be used on all requests to specified clustered hash table server.

It is recommended to disconnect using the "Disconnect Clustered Hash Table (QcstDisconnectCHT) API" on page 172 API when the connection is no longer needed.

Restrictions:

- When this API is called, the clustered hash table server must be active on the requesting node.
- Authority to the specified clustered hash table server is controlled by the authorization list specified on the STRCHTSVR command. If no authorization list was specified then any user is allowed to connect to the specified server. If an authorization list was specified *USE authority is needed to connect to the specified server.
- All other clustered hash table APIs must be run under the same job that issues this API.

Authorities and Locks

Authority to clustered hash table server (if authorization list specified)
*USE

Required Parameter Group

Connection handle
OUTPUT; CHAR(16)

The connection handle to use on all requests into the specified clustered hash table server.

Clustered hash table server name
INPUT; CHAR(10)

The name of the clustered hash table server to connect to.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3C1E E	Required parameter &1 omitted.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB4D E	Cluster Resource Services cannot process the request.
CPFBD09 E	Clustered hash table server &1 not active or not responding.
CPFBD0A E	Clustered hash table server &1 internal error.

API Introduced: V5R2

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

---

Disconnect Clustered Hash Table (QcstDisconnectCHT) API

Required Parameter Group:

1	Connection handle	Input	Char(16)
2	Error code	I/O	Void(*)

Service Program: QCSTCHT
Default Public Authority: *USE
Threadsafe: Yes

The Disconnect Clustered Hash Table (QcstDisconnectCHT) API disconnects the using job from the clustered hash table server. After this API runs the clustered hash table server will not allow any more requests with the specified connection handle.

Restrictions:

- A connection must have been established with the clustered hash table server.
- When this API is called, the clustered hash table server must be active on the requesting node.

- This API must be run under the same job that issued the "Connect Clustered Hash Table (QcstConnectCHT) API" on page 171 API.

Authorities and Locks

Required Parameter Group

Connection handle
INPUT; CHAR(16)

This is an active connection handle for the clustered hash table server. It is obtained by calling the "Connect Clustered Hash Table (QcstConnectCHT) API" on page 171 API.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Error Messages

Message ID	Error Message Text
CPF3C1E E	Required parameter &1 omitted.
CPF3C3C E	Value for parameter &1 not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB4D E	Cluster Resource Services cannot process the request.
CPFBD09 E	Clustered hash table server &1 not active or not responding.
CPFBD0A E	Clustered hash table server &1 internal error.
CPFBD0B E	Connection handle not active.

API introduced: V5R2

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

---

Generate Clustered Hash Table Key (QcstGenerateCHTKey) API

Required Parameter Group:

1	Key	Output	Char(*)
3	Length of key to generate	Input	Binary(4)
2	Connection handle	Input	Char(16)
4	Error Code	I/O	Void(*)

Service Program: QCSTCHT
Default Public Authority: *USE
Threadsafe: Yes

The Generate Clustered Hash Table Key (QcstGenerateCHTKey) API returns a universally unique key that can be used to store an entry into the clustered hash table.

Restrictions:

- A connection must have been established with the clustered hash table server.
- When this API is called, the clustered hash table server must be active on the requesting node.

- This API must be run under the same job that issued the "Connect Clustered Hash Table (QcstConnectCHT) API" on page 171 API.

Authorities and Locks

Required Parameter Group

Key OUTPUT; CHAR(*)

The receiver variable that receives the generated key. The size of this parameter must be at least equal to the length of key to generate parameter.

Length of key to generate
INPUT; BINARY(4)

The length of the key to be generated. This length must be equal to 16.

Connection handle
INPUT; CHAR(16)

This is an active connection handle for the clustered hash table server. It is obtained by calling the "Connect Clustered Hash Table (QcstConnectCHT) API" on page 171 API.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Error Messages

Message ID	Error Message Text
CPF3C1D E	Length specified in parameter &1 not valid.
CPF3C19 E	Error occured with receiver variable specified.
CPF3C1E E	Required parameter &1 omitted.
CPF3C3C E	Value for parameter &1 not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBD09 E	Clustered hash table server &1 not active or not responding.
CPFBD0B E	Connection handle not active.

API introduced: V5R2

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

---

List Clustered Hash Table Keys (QcstListCHTKeys) API

Required Parameter Group:

1	Qualified user space name	Input	Char(20)
2	Format name	Input	Char(8)
3	Connection handle	Input	Char(16)
4	Key selection information	Input	Void(*)
5	Key selection information size	Input	Binary(4)
6	Key selection information format	Input	Char(8)
7	Error code	I/O	Void(*)

Service Program: QCSTCHT
Default Public Authority: *USE
Threadsafe: Yes

The List Clustered Hash Table Keys (QcstListCHTKeys) API generates a list of keys and descriptive information about the entries stored in the clustered hash table specified by the connection handle parameter. The generated list is placed in the specified user space and replaces any existing list. There is no special authority needed to retrieve a list of the keys. The list can include some of the following:

- All entries
- Entries which are in conflict between clustered hash table domain nodes
- Entries which are owned by a specific user
- Entries which were last stored by a specific user

Restrictions:

- A connection must have been established with the clustered hash table server.
- When this API is called, the clustered hash table server must be active on the requesting node.
- This API must be run under the same job that issued the "Connect Clustered Hash Table (QcstConnectCHT) API" on page 171 API.

Authorities and Locks

User Space Authority
*CHANGE

User Space Library Authority
*EXECUTE

User Space Lock
*EXCLRD

Required Parameter Group

Qualified user space name
INPUT; CHAR(20)

The name of the *USRSPC object that is to receive the generated list. The first 10 characters contain the user space object name, and the second 10 characters contain the name of the library where the user space is located. No special values are supported for library name, for example, QTEMP, *CURLIB, or *LIBL. The user space cannot be in an independent auxiliary storage pool.

Format name
INPUT; CHAR(8)

The format of the information returned for each key in the clustered hash table. Possible values are:

CHTL0100	Clustered Hash Table Keys

For more information, see "CHTL0100 Format" on page 177.

Connection handle
INPUT; CHAR(10)

This is an active connection handle for the clustered hash table server. It is obtained by calling the "Connect Clustered Hash Table (QcstConnectCHT) API" on page 171 API.

Key selection information
INPUT; CHAR(*)

The information that determines the keys to be listed. The format of this information is described in "CHTI0100 Format" on page 177.

Key selection information size
INPUT; BINARY(4)

The size in bytes of the key selection information parameter. If the size is greater than the length of the key selection information format, it must be padded with zeroes.

Key selection information format
INPUT; CHAR(8)

The format of the key selection information parameter. Possible values are:

CHTI0100 The specific information identifying the keys to be listed.

For more information, see "CHTI0100 Format" on page 177.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Format of the Generated Lists

The clustered hash table key list consists of:

- A user space
- A generic header
- An input parameter section
- A header section
- A list data section:
  - CHTL0100 format

For details about the user area and generic header, see User space format for list APIs. For detailed descriptions of the fields in the list returned, see "Field Descriptions" on page 178.

The completion code in the generic header should be checked to determine if the API completed successfully. When you retrieve list entry information from a user space, you must use the length of entry information returned associated with each entry returned. If you do not use the length of entry information the result may not be valid.

Input Parameter Section

An exact copy of the parameters coded in the call to the API.

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	CHAR(10)	User space name
10	A	CHAR(10)	User space library name
20	14	CHAR(8)	Format name
28	1C	CHAR(16)	Connection handle
44	2C	CHAR(8)	Key selection information format
52	34	BINARY(4)	Key selection information size
56	38	BINARY(4)	Entry status requested

Offset		Type	Field
Dec	Hex		
60	3C	CHAR(10)	Storing user profile requested
70	46	CHAR(10)	Owning user profile requested

Header Section

Global information about the clustered hash table keys.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Authorization list name
10	A	CHAR(2)	Reserved

CHTL0100 Format

General information about the clustered hash table keys. Detailed information about a key can be obtained by using the "Retrieve Clustered Hash Table Entry (QcstRetrieveCHTEntry) API" on page 179 API.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of entry information returned
4	4	BINARY(4)	Offset to key
8	8	BINARY(4)	Length of key
12	C	BINARY(4)	Entry status
16	10	BINARY(4)	Authority access level
20	14	CHAR(10)	Storing user profile
30	1E	CHAR(10)	Owning user profile
40	28	CHAR(*)	Key

CHTI0100 Format

The following table shows the format of the key selection information parameter for the CHTI0100 format. For detailed descriptions of the fields in the table, see "Field Descriptions" on page 178.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Entry status requested
4	4	CHAR(10)	Storing user profile requested
14	E	CHAR(10)	Owning user profile requested

Field Descriptions

Authority access level. This field describes who is allowed to retrieve and update the entry associated with this key. Valid special values are:

0 The user profile that owns the entry and a user with *ALLOBJ authority is allowed to retrieve and update the entry associated with this key.
1 Any user can retrieve and update the entry associated with this key.

Authorization list name. This is the name of the authorization list specified when the cluster hash table server was started. Valid special values are:

**NONE* No authorization list was specified.

Connection handle. This is an active connection handle for the clustered hash table server. It is obtained by calling the "Connect Clustered Hash Table (QcstConnectCHT) API" on page 171 API.

Entry status. Indicates whether the entry is in conflict or not. An entry is in conflict if it is not the same on all nodes in the clustered hash table domain. A potential cause of an entry marked in conflict is the clustered hash table domain nodes were not communicating and the information associated with the key was updated from more than one cluster partition. For additional details on cluster node partitions, see Partition errors. To resolve an entry in conflict, use the "Store Clustered Hash Table Entry (QcstStoreCHTEntry) API" on page 182 API to update the entry to the correct value. The possible values are:

0 Entry is not in conflict in the clustered hash table.
1 Entry is in conflict in the clustered hash table.

Entry status requested. Specifies the entry status to use when returning the list of keys. The possible values are:

0 Entries which are not in conflict will be returned.
1 Entries which are in conflict will be returned.
-1 All entries will be returned regardless of status. This is the default value.

Format name. The content and format of the information returned for each clustered hash table key. The value must be set to CHTL0100.

Key. A key stored in the clustered hash table.

Length of key. The length of the key stored in the clustered hash table.

Length of entry information returned. The total length of the entry information returned for the key. This value is used to increment to the next key in the list.

Offset to key. The bytes from the beginning of the entry to the Key field.

Owning user profile. The user profile that owns the entry associated with this key.

Owning user profile requested. Specifies the owning user profile to use when returning the list of keys. The owning user profile is the user profile that originally stored the entry. Valid special values for this field are:

*ALL Returns the keys in the clustered hash table for all owners. This special value must be left justified. This is the default value.

Reserved. This field will contain hexadecimal zeroes.

Storing user profile. The user profile that last stored the entry associated with this key.

Storing user profile requested. Specifies the storing user profile to use when returning the list of keys. The storing user profile is the user profile that last stored the entry. Valid special values for this field are:

*ALL Returns the keys in the clustered hash table for all user profiles. This special value must be left justified. This is the default value.

User space library name. The name of the library that contains the user space.

User space name. The name of the user space that receives the list.

Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3C21 E	Format name &1 is not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF9810 E	Library &1 not found.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB4D E	Cluster Resource Services cannot process the request.
CPFBD09 E	Clustered hash table server &1 not active or not responding.
CPFBD0A E	Clustered hash table server &1 internal error.
CPFBD0B E	Connection handle not active.

API introduced: V5R2

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

---

Retrieve Clustered Hash Table Entry (QcstRetrieveCHTEntry) API

Required Parameter Group:

1	Receiver variable	Output	Void(*)
2	Length of receiver variable	Input	Binary(4)
3	Connection handle	Input	Char(16)
4	Format name	Input	Char(8)
5	Length of key	Input	Binary(4)
6	Key	Input	Char(*)
7	Error code	I/O	Void(*)

Service Program: QCSTCHT
Default Public Authority: *USE
Threadsafe: Yes

The Retrieve Clustered Hash Table Entry (QcstRetrieveCHTEntry) API retrieves an entry from the clustered hash table specified by the connection handle parameter. The entry to be retrieved is identified by the key parameter. If the entry exists, is not expired and the requesting user is authorized, the information will be returned in the receiver parameter in the selected format.

Restrictions:

- A connection must have been established with the clustered hash table server.
- When this API is called, the clustered hash table server must be active on the requesting node.
- This API must be run under the same job that issued the "Connect Clustered Hash Table (QcstConnectCHT) API" on page 171 (QcstConnectCHT) API.

Authorities and Locks

Entry Authority Access Level (for restricted entries)
*ALLOBJ or owner of the entry

Required Parameter Group

Receiver variable
OUTPUT; CHAR(*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length of the parameter correctly. As a result, the API returns only the information that the area can hold.

Length of receiver variable
INPUT; BINARY(4)

The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of the receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

Connection handle
INPUT; CHAR(16)

This is an active connection handle for the clustered hash table server. It is obtained by calling the "Connect Clustered Hash Table (QcstConnectCHT) API" on page 171 (QcstConnectCHT) API.

Format name
INPUT; CHAR(8)

The content and format of the information that is returned. The possible format names are as follows:

CHTR0100	Entry information. For more information, see "CHTR0100 Format" on page 181.

Length of key
INPUT; BINARY(4)

The length of the key of the entry to be retrieved. This must be 16 bytes.

Key INPUT; Char(*)

The key of the entry to be retrieved.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

CHTR0100 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Offset to data
12	C	BINARY(4)	Length of data
16	10	BINARY(4)	Entry status
20	14	BINARY(4)	Authority access level
24	18	CHAR(10)	Owning user profile
34	22	CHAR(10)	Storing user profile
44	2C	CHAR(*)	Data

Field Descriptions

Authority access level. This field describes who is allowed to retrieve and update the entry. Valid special values are:

0 The user profile that owns the entry and a user with *ALLOBJ authority is allowed to retrieve and update the entry.
1 Any user can retrieve and update the entry associated with this key.

Bytes available. The number of bytes of information available to be returned to the user.

Bytes returned. The number of bytes of information returned to the user.

Data. The data associated with the specified key.

Entry status. Indicates whether the entry is in conflict or not. An entry is in conflict if it is not the same on all nodes in the clustered hash table domain. A potential cause of an entry in conflict is the clustered hash table domain nodes were not communicating and the information associated with the key was updated from more than one cluster partition. The entry was marked in conflict when the cluster partition was merged together. For additional details on cluster node partitions, see Partition errors. The entry is not the same on all nodes in the clustered hash table domain. To resolve an entry in conflict, use the “Store Clustered Hash Table Entry (QcstStoreCHTEntry) API” on page 182 (QcstStoreCHTEntry) API to update the entry to the correct value. The possible values are:

0 Entry is not in conflict in the clustered hash table.
1 Entry is in conflict in the clustered hash table.

Length of data. Length of the data associated with the specified key.

Offset to data. The offset from the beginning of the structure to the Data field.

Owning user profile. The user profile that originally stored the entry associated with this key.

Storing user profile. The user profile that last stored the entry associated with this key.

Error Messages

Message ID	Error Message Text
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of receiver variable not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB4D E	Cluster Resource Services cannot process the request.
CPFBD06 E	Key not found in clustered hash table &1.
CPFBD07 E	User profile &1 not authorized to clustered hash table entry.
CPFBD09 E	Clustered hash table server &1 not active or not responding.
CPFBD0A E	Clustered hash table server &1 internal error.
CPFBD0B E	Connection handle not active.

API introduced: V5R2

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

---

Store Clustered Hash Table Entry (QcstStoreCHTEntry) API

Required Parameter Group:

1	Connection handle	Input	Char(16)
2	Store description	Input	Void(*)
3	Format name	Input	Char(8)
4	Error code	I/O	Void(*)

Service Program: QCSTCHT
Default Public Authority: *USE
Threadsafe: Yes

The Store Clustered Hash Table Entry (QcstStoreCHTEntry) API stores an entry in the clustered hash table identified by the connection handle.

The storage for the entry is not persistent. Not persistent means the storage for the entry is only known to the clustered hash table server on the local node and only available until the clustered hash table server is ended.

This request to store an entry is replicated to other nodes in the clustered hash table domain. The clustered hash table domain was defined using the STRCHTSVR command. Control will not be returned until the entry is stored in the clustered hash table on all active nodes in the clustered hash table domain.

There is no encrypting of the information that is replicated and stored in the clustered hash table.

When an entry is stored, a time to live value is specified. The entry can become expired, when the time to live value has expired. Expired entries will be removed when processing various functions. For example, when adding another cluster node to the domain of an existing clustered hash table server. The existing entries, if any, are replicated to the added cluster hash table domain node. Expired entries are removed from the clustered hash table during this process.

The user that originally stores the entry will be the owner of the entry. The owning user profile will be used in determining authorization to an entry.

Information stored in the clustered hash table is associated with a key. The key can be generated using the "Generate Clustered Hash Table Key (QcstGenerateCHTKey) API" on page 173 API or the user can generate their own.

An entry in the clustered hash table can be stored with restricted access. This provides the ability to restrict who is allowed to retrieve and update an entry. See authority access level field description for details on entry level authority.

Duplicate keys are not supported. An entry associated with an existing key can be updated if the requesting user is the owner of the entry or is authorized to the entry. See authority access level field description for details on entry level authority.

Restrictions:

- A connection must have been established with the clustered hash table server.
- When this API is called, the clustered hash table server must be active on the requesting node.
- A partition can occur when communication is lost between the cluster nodes defined in the clustered hash table domain. For additional details on cluster node partitions, see Partition errors. The following are the recommendations if the clustered hash table domain is partitioned:
  - Updating the entry associated with an existing key should be restricted to one cluster partition. When the cluster version is 3 or greater, conflicts in the entry found when the cluster merges partitions will be resolved by selecting the entry from the clustered hash table domain node that was last updated. If it is indeterminate which clustered hash table domain node last updated the entry, all clustered hash table domain nodes will mark the entry in conflict. To resolve an entry in conflict use store to update the entry to the correct information.
  - Unique keys can be added from any cluster partition. However, Cluster Resource Services does not guarantee keys are unique between cluster partition. Managing unique keys across cluster partitions is the users responsibility.
  - When the current cluster version is 2 and merging cluster partitions, conflicts in the entry are not checked. Only unique keys between the cluster partitions will be replicated. For more information on the current cluster version see "Cluster Version" on page 224.
- This API must be run under the same job that issued the "Connect Clustered Hash Table (QcstConnectCHT) API" on page 171 API.

Authorities and Locks

Authority to update existing entry (for restricted entries)
*ALLOBJ or owner of the entry

Required Parameter Group

Connection handle
INPUT; CHAR(16)

This is an active connection handle for the clustered hash table server. It is obtained by calling the "Connect Clustered Hash Table (QcstConnectCHT) API" on page 171 API.

Store entry description
INPUT; CHAR(*)

Detailed information for the store request. For more information, see "CHTS0100 Format" on page 184.

Format name
INPUT; Char(8)

The content and format of the information that is stored. The possible format names are as follows:

CHTS0100
Entry description. For more information, see "CHTS0100 Format."

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

CHTS0100 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Offset to key
4	4	BINARY(4)	Length of key
8	8	BINARY(4)	Offset to data
12	C	BINARY(4)	Length of data
16	10	BINARY(4)	Offset to additional fields
20	14	BINARY(4)	Length of additional fields
24	18	BINARY(4)	Update option
28	1C	BINARY(4)	Authority access level
32	20	BINARY(4)	Time to live
*	*	CHAR(*)	Key
*	*	CHAR(*)	Data

Field Descriptions

Authority access level. This field describes who is allowed to retrieve and update the entry associated with this key. This field must be 0 if the current cluster version is 2. For more information about the current cluster version, see "Cluster Version" on page 224. Valid special values are:

0 The owning user profile or a user with *ALLOBJ authority is allowed to retrieve and update the entry.
1 Any user can retrieve and update the entry associated with this key.

Data. The data to be associated with the specified key.

Key. The key that will be associated with the entry. A unique key can be generated by using the "Generate Clustered Hash Table Key (QcstGenerateCHTKey) API" on page 173 API. If the key already exists, the update option field determines the action.

Length of key. The length in bytes of the Key field. This value must be 16.

Length of data. The length in bytes of the Data field. This length can be from 1 to 61000.

Length of additional fields. The length in bytes of the additional fields. This field must be zero.

Offset to data. The bytes from the beginning of this parameter to the Data field.

Offset to key. The bytes from the beginning of this parameter to the Key field.

Offset to additional fields. The bytes from the beginning of this parameter to the Additional fields. This field must be zero.

Time to live. The time (in minutes) that the entry will be allowed to remain in the clustered hash table before expiring. This value can be 1 to 525600 (minutes in one year). A value of -1 can be specified to indicate the entry will never expire.

Update option. The action used when the specified key already exists. This field must be 0 if the current cluster version is 2. For more information about the current cluster version, see "Cluster Version" on page 224. This value is only for the given store request. Valid special values for this field are:

0 Do not allow updating the entry associated with the specified key. If the specified key already exists in the clustered hash table an error will be issued. The behavior for this option will vary if the current cluster version is 2. The uniqueness of the key may not always be detected. To ensure uniqueness use the "Generate Clustered Hash Table Key (QcstGenerateCHTKey) API" on page 173 API to generate a unique key for each store request when the current cluster version is 2.

1 Allow updating the entry associated with the specified key if the specified key already exists and the requesting user is authorized to update the information. See the authority access level field for details on authorization to update an entry.

Error Messages

Message ID	Error Message Text
CPF3C1D E	Length specified in parameter &1 not valid.
CPF3C19 E	Error occured with receiver variable specified.
CPF3C1E E	Required parameter &1 omitted.
CPF3C3C E	Value for parameter &1 not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPFBB26 E	Cluster Resource Services not active or not responding.
CPFBB4D E	Cluster Resource Services cannot process the request.
CPFBB5F E	Field value within structure is not valid.
CPFBB70 E	Request &1 not compatible with current cluster version.
CPFBD07 E	User profile &1 not authorized to clustered hash table entry.
CPFBD08 E	Key already exists in clustered hash table &1.
CPFBD09 E	Clustered hash table server &1 not active or not responding.
CPFBD0A E	Clustered hash table server &1 internal error.
CPFBD0B E	Connection handle not active.

API introduced: V5R2

Top | "Cluster APIs—Introduction" on page 218 | APIs by category

Exit Programs

These are the Exit Programs for this category.

Cluster Resource Group Exit Program

Required Parameter Group:

1	Success indicator	Output	Binary(4)
2	Action code	Input	Binary(4)
3	Exit program data	Input	Char(256)
4	Information given to user	Input	Char(*)

5	Format name	Input	Char(8)

Include: QSYSINC/QCSTCRG3

For each cluster resource group that has an exit program specified, the exit program is called when various "Cluster APIs," on page 1 are used or when various cluster events occur. The exit program is called on each active node in the cluster resource group's recovery domain and is passed an **Action Code** that tells the exit program what function to perform.

An active node in the cluster resource group's recovery domain means that cluster resource services and the job for the particular cluster resource group are running on the node.

The exit program is required for data, application » and peer « cluster resource groups and is responsible for providing and managing the environment necessary for the resource's resilience.

The exit program is optional for device cluster resource groups because the system manages resilient devices. An exit program may be specified for a device cluster resource group if a user has additional functions to perform during various APIs or cluster events.

The exit program is called from a separate job which is started with the user profile specified on the "Create Cluster Resource Group (QcstCreateClusterResourceGroup) API" on page 107 (QcstCreateClusterResourceGroup) API. For most action codes, Cluster Resource Services waits for the exit program to finish before continuing. A time out is not used. If the exit program goes into a long wait such as waiting for a response to a message sent to an operator, no other work will be started for the affected cluster resource group. In the case of a long wait during failover processing for a node failure, all Cluster Resource Services jobs are affected and no other cluster work will be started. Care should be exercised in the exit program when the possibility of a long wait exists.

In general if the exit program is unsuccessful or ends abnormally, the exit program will be called a second time with an action code of Undo. This allows any unfinished activity to be backed out and the original state of the cluster resource group and the resilient resource to be restored. There are some exceptions to this general statement about Undo. Some APIs continue even if the exit program is not successful and do not make a second call with an Undo action code. Also, an application cluster resource group exit program is not called with Undo if it fails while processing the Start action code for a Switchover or Failover.

» Suppose an unsuccessful switchover occurs for a device CRG, after switching back all the devices, if all of the devices are varied-on successfully on the original primary node, clustering calls the exit program on the original primary node with an action code of Start. «

More information on action codes, functions an exit program should perform, and what causes an exit program to be called is presented after the exit program parameters are described.

The exit program is restricted to the Cluster Resource Services APIs or commands it can use. Only the following are allowed:

- "Distribute Information (QcstDistributeInformation) API" on page 127 (QcstDistributeInformation) API
- "List Cluster Information (QcstListClusterInfo) API" on page 37 (QcstListClusterInfo) API
- "List Cluster Resource Group Information (QcstListClusterResourceGroupIn) API" on page 141 (QcstListClusterResourceGroupIn) API
- "List Cluster Resource Groups (QcstListClusterResourceGroups) API" on page 152 (QcstListClusterResourceGroups) API
- "List Device Domain Information (QcstListDeviceDomainInfo) API" on page 41 (QcstListDeviceDomainInfo) API
- "Retrieve Cluster Information (QcstRetrieveClusterInfo) API" on page 50 (QcstRetrieveClusterInfo) API

- "Retrieve Cluster Resource Services Information (QcstRetrieveCRSInfo) API" on page 53 (QcstRetrieveCRSInfo) API
- DSPCLUINF Command
- DSPCRGINF Command

Also, the exit program must follow these guidelines to run properly in the job Cluster Resource Services starts for it and to handle error conditions correctly.

- The exit program cannot be in an **independent** auxiliary storage pool.
- It must run in a named activation group or the caller's activation group (*CALLER), when the cluster resource group exit program is an ILE program.
- It must have a cancel handler to deal with situations where the job is cancelled. In particular, an application cluster resource group exit program will have its job cancelled as part of the "Initiate Switchover (QcstInitiateSwitchOver) API" on page 135 (QcstInitiateSwitchOver) API.
- It should have an exception handler to deal with unexpected exceptions and perform as much cleanup as possible at that point while it can still interrogate program state information.
- It may use threads but the job description in the user profile for the job must specify that threads are allowed.
- » The user profile that the cluster resource group exit program runs under must not have an IASP associated with the user profile job description. «

Note: See "Cluster Resource Services Job Structure" on page 220 for additional information about jobs used to call exit programs.

» High Availability Business Partners (HABPs) « provide software products that replicate data to other nodes in a cluster by using data cluster resource groups. Application cluster resource groups may have dependencies on these data cluster resource groups. An application cluster resource group exit program can be used to coordinate activities with data cluster resource group exit programs that are provided by an HABP.

Sample source code that can be used as the basis for writing an exit program is shipped in the QUSRTOOL library. See the TCSTAPPEXT and TCSTDTAARA members in the QATTSYSC file for an example written in ILE C.

Authorities and Locks

None.

Required Parameter Group

Success indicator
OUTPUT; BINARY(4)

Indicates to Cluster Resource Services the results of the cluster resource group exit program. The exit program must set this parameter before it ends. If the job running the exit program is cancelled before the exit program ends, the exit program cancel handler should set this parameter. Possible values of this parameter

0 Successful.
1 Unsuccessful, do not attempt restart.
2 Unsuccessful, attempt restart (applies only to an application cluster resource group).

Some APIs ignore this field. In other words, regardless of what value is set by the exit program these functions continue to completion and do not backout partial results or call the exit program a second time with an Undo action code. Likewise, the exit program should make every attempt to complete successfully for these APIs. This field is ignored by the following:

- "Change Cluster Node Entry (QcstChangeClusterNodeEntry) API" on page 14 (QcstChangeClusterNodeEntry) API
- Change Cluster Node Entry (CHGCLUNODE) Command
- "Delete Cluster (QcstDeleteCluster) API" on page 31 (QcstDeleteCluster) API
- Delete Cluster (DLTCLU) Command
- "Delete Cluster Resource Group (QcstDeleteClusterResourceGroup) API" on page 124 (QcstDeleteClusterResourceGroup) API for the Delete action code
- "Delete Cluster Resource Group (QcstDeleteClusterResourceGroup) API" on page 124 for the Delete action code
- Delete Cluster Resource Group (DLTCRG) command.
- "End Cluster Node (QcstEndClusterNode) API" on page 33 (QcstEndClusterNode) API
- "Remove Cluster Node Entry (QcstRemoveClusterNodeEntry) API" on page 44 (QcstRemoveClusterNodeEntry) API
- End Cluster Node (ENDCLUNOD) command
- Remove Cluster Node Entry (RMVCLUNODE) command
- Failover Cancelled action code

An informational, alert message, CPIBB10, will be sent if the exit program returns anything other than Successful, has an unhandled exception, or the job running the exit program is cancelled.-

See "When the Exit Program Ends" on page 204 for additional information on the Success indicator.

Action code

INPUT; BINARY(4)

Identifies the cluster API or event that is being processed and, therefore, the action the exit program should perform. The action codes listed below apply to all cluster resource group types unless otherwise specified. See also action code dependent data field in the "Field Descriptions" on page 193 below. This further defines the action code.

Possible action codes:

1 - Initialize
A cluster resource group object is being created.

2 - Start
Establish resilience for a cluster resource group. In addition, start the application for an application cluster resource group.

The exit program job for an application cluster resource group remains active on the primary node. Thus when a "Cluster Resource Group APIs" on page 61 is called while an application is running, a second job will be submitted on the primary node to run the exit program to handle the API action code.

» For a peer cluster resource group all peer nodes will be started and the exit program will be called. «

» Suppose an unsuccessful switchover occurs for a device CRG. After switching back all the devices, if all of the devices are varied-on successfully on the original primary node, the exit program is called on the original primary node. «

3 - Restart
Restarts an application. Applies **only** to an application cluster resource group which has been configured to allow restart.

4 - End
Disable resilience of a cluster resource group on all nodes in the recovery domain.

5 - Verification phase
Provides a chance for the cluster resource group exit program to verify it really wants to perform the requested function before actual doing the specified function. This is similar to a pre-exit program.

7 - Delete
The cluster resource group object is being deleted.

8 - Rejoin	An inactive node is being activated or communication with a partitioned node is re-established. If cluster resource services is not running on any node in the cluster, the first node which is started will cause the exit program to be called with an action code of Rejoin. In this case, there will be no other active nodes in the cluster.
9 - Failover	A node failure has occurred.
10 - Switchover	Either the "Initiate Switchover (QcstInitiateSwitchOver) API" on page 135 or the Change Cluster Resource Group Primary (CHGCRGPRI) command is being processed. The role of primary is being assigned to the node that has the current role of first backup. » Does not apply to peer cluster resource groups. «
11 - Add Node	A node is being added to the recovery domain. » If the cluster resource group is active and a peer node is being added to a peer cluster resource group, the node will be an active access point. «
12 - Remove Node	A node is being removed from the recovery domain. When an active node is being removed with either the "Remove Cluster Node Entry (QcstRemoveClusterNodeEntry) API" on page 44 or the Remove Cluster Node Entry (RMVCLUNODE) command, only the node being removed sees this action code. Other nodes in the recovery domain will see the Failover action code. When an inactive node is being removed with either the "Remove Cluster Node Entry (QcstRemoveClusterNodeEntry) API" on page 44 or the Remove Cluster Node Entry (RMVCLUNODE) command, all nodes see this action code.
13 - Change	The nodes listed in the recovery domain have changed their role. » For a peer cluster resource group all the recovery domain nodes are provided even if only some of them have their role changed. «
14 - Delete Command	The cluster resource group is being deleted on the local node only.
15 - Undo	Backout the prior request. The prior request is in the **Prior Action Code** field.
16 - End Node	A cluster node is ending. Only the node being ended will see this action code. Other nodes in the recovery domain will see the Failover action code.
17 - Add Device Entry	A device is being added to a device cluster resource group. Applies **only** to a device cluster resource group.
18 - Remove Device Entry	A device is being removed from a device cluster resource group. Applies **only** to a device cluster resource group.
19 - Change Device Entry	Information about a device in a device cluster resource group is being changed. Applies **only** to a device cluster resource group.
20 - Change Node Status	The status of a node is being changed from partitioned to failed.
21 - Failover Cancelled	A node failure occurred, but the failover was cancelled through the use of the failover message queue.

Exit program data

INPUT; CHAR(256)

Because this parameter is passed between nodes in the cluster, it can contain anything **except** pointers. For example, it can be used to provide state information. The owner of the cluster resource group knows the layout of the information contained in this parameter.

This data comes into existence when the cluster resource group object is created with the "Create Cluster Resource Group (QcstCreateClusterResourceGroup) API" on page 107 or the Create Cluster Resource Group (CRTCRG) command. Change this data in the following ways:

- "Change Cluster Resource Group (QcstChangeClusterResourceGroup) API" on page 86 or Change Cluster Resource Group (CHGCRG) command
- "End Cluster Resource Group (QcstEndClusterResourceGroup) API" on page 132 or End Cluster Resource Group (ENDCRG) command
- "Start Cluster Resource Group (QcstStartClusterResourceGroup) API" on page 164 or Start Cluster Resource Group (STRCRG) command
- "Initiate Switchover (QcstInitiateSwitchOver) API" on page 135 or Change Cluster Resource Group Primary (CHGCRGPRI) command

See the description of each API in "Cluster Resource Group APIs" on page 61.

Information given to user
INPUT; CHAR(*)

Detailed information for this exit program call. See the "EXTP0100 Format" or "EXTP0200 Format" on page 191 for more information.

Format name
INPUT; CHAR(8)

The format of the information provided in the Information Given To User parameter. If the exit program is called with a second action code such as Undo, the format contains the same data as was passed the original action code. The format name supported is:

EXTP0100 This format is used for actions. This value is allowed for primary-backup model cluster resource group and peer model cluster resource group.

EXTP0200 Same as EXTP0100, with additional information on site name and data port IP addresses on each node in the recovery domain. This value is allowed for primary-backup model cluster resource group.

EXTP0100 Format

This format contains information for the cluster event .

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of information given to user
4	4	CHAR(10)	Cluster name
14	E	CHAR(10)	Cluster resource group name
24	18	BINARY(4)	Cluster resource group type
28	1C	BINARY(4)	Cluster resource group status
32	20	CHAR(16)	Request handle
48	30	BINARY(4)	Node role type
52	34	CHAR(8)	Current node id
60	3C	CHAR(8)	Changing node ID
68	44	BINARY(4)	Changing node role
72	48	CHAR(16)	Takeover IP address
88	58	CHAR(10)	Job name
98	62	CHAR(2)	Reserved
100	64	BINARY(4)	Prior action code
104	68	BINARY(8)	Cluster resource group changes
112	70	BINARY(4)	Offset to recovery domain array
116	74	BINARY(4)	Number of nodes in the recovery domain
120	78	BINARY(4)	Original cluster resource group status
124	7C	BINARY(4)	Action code dependent data
128	80	BINARY(4)	Offset to prior recovery domain array
132	84	BINARY(4)	Number of nodes in the prior recovery domain array
136	88	BINARY(4)	Offset to configuration object array
140	8C	BINARY(4)	Number of entries in configuration object array
144	90	BINARY(4)	Length of configuration object array entry

Offset		Type	Field
Dec	Hex		
148	94	BINARY(8)	Cluster resource group attributes
156	9C	CHAR(10)	Distribute information user queue name
166	A6	CHAR(10)	Distribute information user queue library name
176	B0	BINARY(4)	Failover wait time
180	B4	BINARY(4)	Failover default action
184	B8	CHAR(10)	Failover message queue name
194	C2	CHAR(10)	Failover message queue library name
204	CC	BINARY(4)	Cluster version
208	D0	BINARY(4)	Cluster version modification level
212	D4	CHAR(10)	Requesting user profile
» 222	DE	CHAR(1)	Reserved «
» 222	DF	CHAR(1)	Allow active takeover IP address «
» 224	E0	CHAR(20)	Application id «
» 244	F4	BINARY(4)	Length of recovery domain array entry «
» 248	F8	BINARY(4)	Length of prior recovery domain array entry «
» 252	FC	CHAR(8)	Leader node id «
*	*	Array(*) of CHAR(*)	Recovery domain array
These fields repeat, in the order listed, for each node in the recovery domain.		CHAR(8)	Node ID
		BINARY(4)	Node role
		BINARY(4)	Membership status
*	*	Array(*) of CHAR(*)	Prior recovery domain array
These fields repeat, in the order listed, for each node in the recovery domain.		CHAR(8)	Node ID
		BINARY(4)	Node role
		BINARY(4)	Membership status
*	*	Array(*) of CHAR(44)	Configuration object array
These fields repeat, in the order listed, for each entry in the configuration object array.		CHAR(10)	Configuration object name
		CHAR(2)	Reserved
		BINARY(4)	Configuration object type
		BINARY(4)	Device type
		BINARY(4)	Configuration object online
		BINARY(4)	Device subtype
		CHAR(16)	Server takeover IP address

EXTP0200 Format

This format » contains information for the cluster event with additional information on site name and data port IP addresses on each node in the recovery domain. «

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of information given to user

Offset		Type	Field
Dec	Hex		
4	4	CHAR(10)	Cluster name
14	E	CHAR(10)	Cluster resource group name
24	18	BINARY(4)	Cluster resource group type
28	1C	BINARY(4)	Cluster resource group status
32	20	CHAR(16)	Request handle
48	30	BINARY(4)	Node role type
52	34	CHAR(8)	Current node id
60	3C	CHAR(8)	Changing node ID
68	44	BINARY(4)	Changing node role
72	48	CHAR(16)	Takeover IP address
88	58	CHAR(10)	Job name
98	62	CHAR(2)	Reserved
100	64	BINARY(4)	Prior action code
104	68	BINARY(8)	Cluster resource group changes
112	70	BINARY(4)	Offset to recovery domain array
116	74	BINARY(4)	Number of nodes in the recovery domain
120	78	BINARY(4)	Length of recovery domain array entry
124	7C	BINARY(4)	Original cluster resource group status
128	80	BINARY(4)	Action code dependent data
132	84	BINARY(4)	Offset to prior recovery domain array
136	88	BINARY(4)	Number of nodes in the prior recovery domain array
140	8C	BINARY(4)	Length of prior recovery domain array entry
144	90	BINARY(4)	Offset to configuration object array
148	94	BINARY(4)	Number of entries in configuration object array
152	98	BINARY(4)	Length of configuration object array entry
156	9C	BINARY(8)	Cluster resource group attributes
164	A4	CHAR(10)	Distribute information user queue name
174	AE	CHAR(10)	Distribute information user queue library name
184	B8	BINARY(4)	Failover wait time
188	BC	BINARY(4)	Failover default action
192	C0	CHAR(10)	Failover message queue name
202	CA	CHAR(10)	Failover message queue library name
212	D4	BINARY(4)	Cluster version
216	D8	BINARY(4)	Cluster version modification level
220	DC	CHAR(10)	Requesting user profile
230	E6	CHAR	Reserved
231	E7	CHAR	Reserved
*	*	Array(*) of CHAR(*)	Recovery domain array

Offset		Type	Field
Dec	Hex		
These fields repeat, in the order listed, for each node in the recovery domain.		BINARY(4)	Length of entry in the recovery domain
		CHAR(8)	Node ID
		BINARY(4)	Node role
		BINARY(4)	Membership status
		CHAR(8)	Site name
		BINARY(4)	Offset to data port IP address array
		BINARY(4)	Number of data port IP addresses
		Array(*) of CHAR(16)	Data port IP address
*	*	Array(*) of CHAR(*)	Prior recovery domain array
These fields repeat, in the order listed, for each node in the recovery domain.		BINARY(4)	Length of entry in the recovery domain
		CHAR(8)	Node ID
		BINARY(4)	Node role
		BINARY(4)	Membership status
		CHAR(8)	Site name
		BINARY(4)	Offset to data port IP address array
		BINARY(4)	Number of data port IP addresses
		Array(*) of CHAR(16)	Data port IP address
*	*	Array(*) of CHAR(44)	Configuration object array
These fields repeat, in the order listed, for each entry in the configuration object array.		CHAR(10)	Configuration object name
		CHAR(2)	Reserved
		BINARY(4)	Configuration object type
		BINARY(4)	Device type
		BINARY(4)	Configuration object online
		BINARY(4)	Device subtype
		CHAR(16)	Server takeover IP address

Field Descriptions

Action code dependent data. For some action codes, additional information is provided to describe the action code. This field is used during:

- Delete action code
- End action code
- Failover action code
- Failover Cancelled action code
- Rejoin action code
- Remove Node action code
- Start action code
- Undo action code
- Verification Phase action code

The possible values are:

0 - No information	No additional information.
1 - Merge	Provided with the Rejoin action code to indicate cluster partitions are merging.
2 - Join	Provided with the Rejoin action code to indicate a node which was Inactive is joining the cluster.
3 - Partition failure	Provided with the Failover or End action code to indicate the cluster has become partitioned. When provided with Failover, this is a primary partition. When provided with End, this is a secondary partition.
4 - Node failure	Provided with the Failover or Failover Cancelled action code to indicate Cluster Resource Services has failed for the entire node. This may mean the node failed such as with a power failure or that all of Cluster Resource Services has failed such as when the QSYSWRK subsystem is cancelled but the node is still running.
5 - Member failure	Provided with the Failover, Failover Cancelled, and End Node action codes to indicate a failure affecting only this cluster resource group has been detected. While other cluster resource groups may be affected also, Cluster Resource Services is unable to determine that. An example of a member failure is when a single cluster resource group job is cancelled.
6 - End node	Provided with the Failover or Failover Cancelled action code to indicate an active node in the cluster has been ended by either the "End Cluster Node (QcstEndClusterNode) API" on page 33 or the End Cluster Node (ENDCLUNOD) command.
7 - Remove node	Provided with the Failover action code to indicate an active node in the cluster has been removed by either the "Remove Cluster Node Entry (QcstRemoveClusterNodeEntry) API" on page 44 or the Remove Cluster Node Entry (RMVCLUNODE) command.
8 - Application failure	Provided with the Failover or Failover Cancelled action code to indicate the application failed on the primary node and could not be restarted there. The failure may have been due to an exception in the application program, an Unsuccessful attempt restart success indicator but the restart count has been reached, or an Unsuccessful do not attempt restart success indicator.
9 - Resource end	Provided with the End action code to indicate a resource has ended. This occurs when an application ends normally for an active application cluster resource group or when an active device cluster resource group has a hardware failure of an auxiliary storage pool that prevents failover. Normal ending of an application occurs when the success indicator is set to Successful and the job has not been cancelled or had an unhandled exception.
10 - Delete cluster	Provided with the Delete action code to indicate the cluster is being deleted with either the "Delete Cluster (QcstDeleteCluster) API" on page 31 or the Delete Cluster (DLTCLU) command.
11 - Remove recovery domain node	Provided with the Remove Node action code to indicate the node is being removed with either the "Remove Node From Recovery Domain (QcstRemoveNodeFromRcvyDomain) API" on page 160 or the Remove Cluster Resource Group Node Entry (RMVCRGNODE) command.
12 - Delete cluster resource group	Provided with the Verification phase action code to indicate the cluster resource group is being deleted with the Delete Cluster Resource Group API or commands. The Delete Cluster Resource Group API and Delete Cluster Resource Group from Cluster command can now be rejected if any recovery domain node fails the verification phase(5) action code and will not be called with an Undo action code.
13 - Failover	Provided with the Verification phase action code. This indicates the cluster resource group is being failed over to a backup node.
14 - Switchover	Provided with the Verification phase action code. This indicates the cluster resource group is being failed over to a backup node.
15 - Remove passive node	Provided with the Remove Node and the Delete action code. This indicates a node that is not active is being removed from the cluster and it needs to be removed from the recovery domain. The delete action code is used if the node being removed is the only node in the recovery domain or there are no other nodes in the recovery domain available to assume primary role (i.e. only replicates left), the cluster resource group is being deleted from the cluster.

» **Allow active takeover IP address.** Allows a takeover IP address to already be active when it is assigned to an application cluster resource group. This field is only valid when configure takeover IP address field is 0x01. Possible values are:

0	The takeover IP address must not already be active when starting the cluster resource group.
1	The takeover IP address is allowed to be active prior to starting the cluster resource group but only on the primary node.

«

» **Application id.** This is an application identifier for the Peer CRG type. It identifies the application supplying the peer cluster. Recommend format is 'vendor-id.name' where vendor-id is an identifier for the vendor creating the cluster resource group (i.e.QIBM.ExamplePeer). This indicates it is supplied by IBM for the ExamplePeer application. It is not recommended to use QIBM for vendor id name unless the cluster resource group is supplied by IBM. This field only applies to peer cluster resource groups. «

Changing node ID. The node in the recovery domain being assigned a new role or status. This field is hexadecimal zeroes if it doesn't apply.

A special value of *LIST is specified for this parameter when more than one node is changed. The special value is left-justified. When *LIST is specified, entries in the recovery domain array and the prior recovery domain array can be compared to determine which nodes have had changes to the node role or membership status.

This field is used during:

- Add Node action code
- Change Node Status action code
- Change action code
- End Node action code
- Failover Cancelled action code
- Switchover action code
- Remove Node action code
- Failover or Failover Cancelled action code
- Rejoin action code

Changing node role. The role the node is being assigned. This field is used by the same situations that the Changing node ID field is used. The values are:

0	Primary node. Only one node can have this value.
>=1	Backup node. The backup order is designated by increasing value. The values need not be consecutive. No two backup nodes can have the same value. At the completion of the API, Cluster Resource Services will sequence the backups using consecutive numbers starting with 1.
-1	Replicate node. All replicates have this value.
-2	Changing node role not used by the action code being processed.
-3	*LIST. When *LIST is specified, entries the recovery domain array and the prior recovery domain array can be compared to determine which nodes have had changes to the node role or membership status.
»	Peer node. All peer nodes have this value. «
-4	

Cluster name. The name of the cluster containing the cluster resource group.

Cluster resource group attributes. A bit mask that identifies various cluster resource group attributes. The 64 bits in this field are numbered 0 thru 63 starting with the rightmost bit. If a bit is set to '1', it indicates the cluster resource group has that attribute. The meaning of each of the bits are:

This field applies only to application cluster resource groups.

0	The takeover IP address is configured by the user
1-63	Reserved. These will be set to '0'.

Cluster resource group changes. A bit mask that identifies the fields in the cluster resource group that are being changed by the Change Cluster Resource Group API. Set to hexadecimal zeroes for all other

exit program calls. The 64 bits in this field are numbered 0 thru 63 starting with the rightmost bit. If a bit is set to '1', it indicates that the action represented by the bit is occurring. Even though multiple bits may be set to indicate several things are being changed, the exit program is called only when the recovery domain is changed. For more information, see the "Change Cluster Resource Group (QcstChangeClusterResourceGroup) API" on page 86 or Change Cluster Resource Group (CHGCRG) command. This field is used by the Change and Undo action codes. The meaning of each of the bits is:

0	Recovery domain is changing
1	Takeover IP address is changing
2-63	Reserved. These will be set to '0'.

Cluster resource group name. The cluster resource group that is being processed by Cluster Resource Services.

Cluster resource group status. Status of the cluster resource group at the time the exit program is called. Possible values include:

10 - Active	For Rejoin action code.
20 - Inactive	For action codes of Rejoin or Delete Command.
30 - Indoubt	For Rejoin action code.
40 - Restored	For Rejoin action code.
500 and greater - Pending	Pending values set by various APIs.

Additional information for cluster resource group status can be found in "Cluster Resource Group APIs" on page 61.

Cluster resource group type. The type of cluster resource group:

1	Data resilience
2	Application resilience
3	Device resilience
» 4	Peer resilience «

Cluster version. The exit program is being called to process the action code at this cluster version. This value determines the cluster's ability to use new functions supported by the cluster. It is set when the cluster is created and can be changed by the "Adjust Cluster Version (QcstAdjustClusterVersion) API" on page 11 or Change Cluster Version (CHGCLUVER) command. Note: When the Adjust Cluster Version API is executed, there is a small window of time where the cluster and cluster resource group job may be operating at different cluster versions.

Cluster version modification level. The exit program is being called to process the action code at this modification level The modification level further identifies the version at which the nodes in the cluster can communicate. It is updated when code changes that impact the version are applied to the system. Note: When the Adjust Cluster Version API is executed, there is a small window of time where the cluster and cluster resource group job may be operating at different cluster version modification levels.

Configuration object array. This array identifies the resilient devices that can be switched from one node to another. This array is present only for a device cluster resource group.

Configuration object name. The name of the auxiliary storage pool device description object which can be switched between the nodes in the recovery domain. An auxiliary storage pool device description can be specified in only one cluster resource group.

Configuration object online. Vary the configuration object on or leave the configuration object varied off when a device is switched from one node to another or when it is failed over to a backup node. Possible values are:

0 Do not vary the configuration object on and do not start the server takeover IP address.
1 Vary the configuration object on and start the server takeover IP address.
2 Perform the same action for a seondary auxiliary storage pool as is specified in the primary.

Configuration object type. This specifies the type of configuration object specified with configuration object name. Possible values are:

1 Device description

Current node ID. Identifies the node running the exit program.

Data port IP address. The IP address associated with the recovery domain node. This is a dotted decimal format field and is a null-terminated string.

Device subtype. A device's subtype. This information is only as current as the last time the cluster resource group object could be updated. If configuration changes have been made on the node which owns the hardware and those changes have not yet been distributed to all nodes in the cluster, this information may be inaccurate. The data cannot be distributed if the configuration was changed on a node which does not have cluster resource services running. Possible values are:

-1 The subtype cannot be determined because hardware configuration is not complete.
0 This device type does not have a subtype.
1 UDFS independent auxiliary storage pool.
2 Secondary independent auxiliary storage pool.
3 Primary independent auxiliary storage pool.

Device type. This specifies the type of device. Possible values are:

1 Auxiliary storage pool

Distribute information user queue library name. The name of the library that contains the user queue to receive the distributed information. This field will be set to hexadecimal zeros if no distribute information user queue name was specified when the cluster resource group was created.

Distribute information user queue name. The name of the user queue to receive distributed information from the Distribute Information API. This field will be set to hexadecimal zeros if no distribute information user queue name was specified when the cluster resource group was created.

Failover default action. Should a response to the failover message queue not be received in the failover wait time limit, then this field tells clustering what it should do pertaining to the failover request. This field applies to all » primary-backup model « cluster resource groups.

0 Proceed with failover.
1 Do NOT attempt failover.

Failover message queue library name. The name of the library that contains the user queue to receive failover messages. This field will be set to hexadecimal zeros if no failover response user queue name was specified. This field applies to all » primary-backup model « cluster resource groups.

Failover message queue name. The name of the message queue to receive messages dealing with failover. This field will be set to hexadecimal zeros if no failover response user queue name was specified. This field applies to all » primary-backup model « cluster resource groups.

Failover wait time. Number of minutes to wait for a reply to the failover message that was enqueued on the failover message queue. This field applies to all » primary-backup model « cluster resource groups.

-1 Wait forever until a response is given to the failover inquiry message.
0 Failover proceeds without user intervention. Acts the same as V5R1M0 and prior.
>=1 Number of minutes to wait for a response to the failover inquiry message. If no response is received in the specified number of minutes, the failover default action field will be looked at to decide how to proceed.

Job name. Name of the job associated with an application cluster resource group exit program. This field is used only by application cluster resource groups.

» **Leader node id.** This field identifies the name of a recovery domain node that is actively participating in the current protocol for the given cluster resource group. A value of hexadecimal zero means the exit program cannot use this field. This field only applies to a peer cluster resource group.

The leader node id is available for these action codes:

- Start
- End
- Remove Node (only if removing a node from the recovery domain)
- Change
- Delete (only if deleting the cluster resource group)

«

Length of configuration object array entry. This specifies the length of an entry in the configuration object array. This field applies only to device cluster resource groups.

Length of entry in the recovery domain. The length of an entry in the recovery domain array. This field is used if each entry may have a different length.

Length of prior recovery domain array entry. The length of an entry in the prior recovery domain array. » For EXTP0100 format this length should be used to navigate to the next prior recovery domain array entry. «

Length of recovery domain array entry. The length of an entry in the recovery domain array. » For EXTP0100 format this length should be used to navigate to the next recovery domain array entry. «

Length of information given to user. The length of the data passed in the format.

Membership status. The cluster resource group membership status for the current role of a node:

0 Active. Cluster Resource Services for this cluster resource group is active on the node.
1 Inactive. Cluster Resource Services for this cluster resource group is not active on the node. The node may have failed, the node may have been ended, the QSYSWRK subsystem on that node which runs the Cluster Resource Services jobs may have been ended, or the cluster resource services job on that node may not be running.
2 Partition. The node has become partitioned and Cluster Resource Services cannot determine whether the node is active or inactive.
3 Ineligible. Cluster Resource Services for this cluster resource group is active on the node but not eligible to become the cluster resource group primary node.

Node ID. A unique string of characters that identifies a node in the recovery domain.

Node role. The role a node is to be assigned at the successful completion of the action code being processed. » For primary-backup model cluster resource groups node can have one of three roles: primary, backup, or replicate. For peer model cluster resource groups a node can have one of two roles: peer or replicate. Any number of nodes can be designated as the peer or replicate. «

0 Primary node. Only one node can have this value. » The primary node can become the active access point for the cluster resource. «

>=1 Backup node. The backup order is designated by increasing value. The values need not be consecutive. No two backup nodes can have the same value. At the completion of the API, Cluster Resource Services will sequence the backups using consecutive numbers starting with 1. » Backup nodes are available to be become active access points for the cluster resource after the primary node. «

-1 Replicate node. All replicates have this value. » Replicate nodes are not ordered and cannot become an access point unless the role is changed to a value appropriate for the cluster resource group type.

-4 Peer node. All peer nodes have this value. Peer nodes are not ordered and can all become active access points at the same time when the cluster resource group is started. «

Node role type. Indicates which of the two node roles is being processed:

1 Current
2 Preferred

Number of entries in configuration object array. The number of resilient device entries in the Configuration Object Entry array. This field has a value of 0 for a data or application cluster resource group. This field applies only to device cluster resource groups.

Number of data port IP addresses. The number of data port IP addresses associated with the recovery domain node. This field has a value of 0 for a data or application cluster resource group. This field applies only to device cluster resource groups.

Number of nodes in the prior recovery domain. The number of nodes in the prior recovery domain. This is the number of elements there are in the Prior Recovery Domain Array. This will be 0 if the Prior Recovery Domain Array is not included. This field is used during:

- Add Node action code
- Change action code
- Change Node Status action code
- Failover action code
- Failover Cancelled action code
- Rejoin action code
- Remove Node action code
- Restart action code
- Start action code
- Switchover action code
- Undo action code

Number of nodes in the recovery domain array. The number of nodes in the recovery domain. This is the number of elements in the recovery domain array.

Offset to configuration object array. The byte offset from the beginning of the format to the list of resilient devices. This field has a value of 0 for a » non-device « cluster resource group. This field applies only to device cluster resource groups.

Offset to data port IP address array. The byte offset from the beginning of the format to the list of data port IP addresses for a recovery domain node. This field has a value of 0 for a » non-device « cluster resource group. This field applies only to device cluster resource groups.

Offset to prior recovery domain array. The byte offset from the beginning of the format to the array of nodes in the prior recovery domain. This will be 0 if the prior recovery domain array is not included. This field is used during:

- Add Node action code
- Change action code
- Change Node Status action code
- Failover action code
- Failover Cancelled action code
- Rejoin action code
- Remove Node action code
- Restart action code
- Start action code
- Switchover action code
- Undo action code

Offset to recovery domain array. The byte offset from the beginning of the format to the array of nodes in the recovery domain.

Original cluster resource group status. The original status of the cluster resource group before it was changed to some pending status while an API is running. For example when the exit program is called for the "Start Cluster Resource Group (QcstStartClusterResourceGroup) API" on page 164, the Cluster resource group status field will contain 550 (Start CRG Pending) while this field will contain 20 (Inactive) or 30 (Indoubt). Possible values include:

10	Active
20	Inactive
30	Indoubt
40	Restored

Additional information for cluster resource group status can be found in "Cluster Resource Group APIs" on page 61.

Preferred node role. The preferred role a node is assigned. » See **Node role** for a more detailed description of the node role. «

0	Primary node. Only one node can have this value.
>=1	Backup node. The backup order is designated by increasing value. The values need not be consecutive. No two backup nodes can have the same value. At the completion of the API, Cluster Resource Services will sequence the backups using consecutive numbers starting with 1.
-1	Replicate node. All replicates have this value. »
-4	Peer node. All peers have this value. Only valid in a peer cluster resource group. «

Prior action code. When a cluster resource group exit program is called with an action code of Undo (15), the action code for the unsuccessful operation is placed in this field. Otherwise, this will be hex zeroes.

Prior recovery domain array. The prior recovery domain array contains the view of the recovery domain before changes were made as a result of the API being used or a cluster event occurring.

For example if a switchover is done, the prior recovery domain array will have the view with the old primary and backup order. The recovery domain array will have the view with the new primary and backup order.

If an event such as a node failure occurs, the prior recovery domain array will have the old membership status for the failing node such as Active while the recovery domain array will have the new status such as Inactive.

In most cases, the prior recovery domain is a view of the current recovery domain. If the "Change Cluster Resource Group (QcstChangeClusterResourceGroup) API" on page 86 is being used to change the preferred recovery domain, the prior recovery domain will have a view of the preferred recovery domain.

The prior recovery domain array is available for these action codes:

- Add Node
- Change
- Change Cluster Node Status
- End Node
- Failover
- Failover Cancelled
- Rejoin
- Remove Node
- Start (Only if inactive backup nodes were reordered in the recovery domain. See "Start Cluster Resource Group (QcstStartClusterResourceGroup) API" on page 164 for more information.)
- Switchover

Recovery domain array. The nodes that are the recovery domain for the cluster resource group. This view of the recovery domain will contain any changes made to the node's membership status or the node's role by the API or cluster event which caused the exit program to be called.

Request handle. Uniquely identifies the API request. It is used to associate responses on the user queue specified in the Results Information parameter. This field will have a null value when the exit program is called with an action code of Failover (9).

Requesting user profile. This is the user profile that initiated the API request.

Reserved. This field is reserved and is set to hexadecimal zeroes.

Server takeover IP address. This is a takeover IP address for servers associated with the relational database. This is a dotted decimal field and is a null-terminated string. » This field only applies to device cluster resource groups. «

Site name. The name of the site associated with the recovery domain node. » This field only applies to device cluster resource groups. «

Takeover IP address. This is the floating IP address that is associated with an application. This is a dotted decimal field and is a null-terminated string. This field is used only by application cluster resource groups.

Application Takeover IP Address Management

The takeover IP address is the IP address used to control how clients access the application as the point of access for the application moves from one node to another during Switchover or failover. The takeover IP address is started only on one node at a time. That node is the primary node in the cluster resource group's recovery domain. The takeover IP address can be configured by Cluster Resource Services or it

can be configured by the user. This attribute is specified on the Create Cluster Resource Group API and is passed to the exit program in the cluster resource group attributes field.

The following table shows which cluster APIs and events configure and manage the takeover IP address. This occurs only for application cluster resource groups. Additional information on the takeover IP address can be found in "Cluster Resource Group APIs" on page 61

Table 1. Takeover IP Address Management

API or cluster event	Cluster Resource Services does Configuration	User does Configuration
Add Node to Recovery Domain API Add Cluster Resource Group Node Entry Command	• Cluster Resource Services ensures the IP address does not exist on the node being added. • Cluster Resource Services adds the IP address to the node being added.	• If the cluster resource group is active and the node being added is a backup node, Cluster Resource Services ensures the IP address is not active on the node being added.
cancel job The exit program job running as a result of handling the Start action code is cancelled by some operator action.	• Cluster Resource Services ends the IP address after the exit program's cancel handler ends.	• Cluster Resource Services ends the IP address after the exit program's cancel handler ends.
Change Cluster Node Entry API Change Cluster Node Entry Command	• Cluster Resource Services does not do anything with the IP address.	• Cluster Resource Services does not do anything with the IP address.
Change Cluster Recovery API Change Cluster Recovery Command	• Cluster Resource Services does not do anything with the IP address.	• Cluster Resource Services does not do anything with the IP address.
Change Cluster Resource Group API Change Cluster Resource Group Command	• When the takeover IP address is being changed, Cluster Resource Services removes the old IP address on all nodes in a cluster resource group's recovery domain and adds the new IP address. • If the cluster resource group is active and the role of a replicate node is being changed to a backup node, Cluster Resource Services ensures the takeover IP address exists and is not active.	• If the cluster resource group is active and the role of a replicate node is being changed to a backup node, Cluster Resource Services ensures the takeover IP address exists and is not active.

API or cluster event	Cluster Resource Services does Configuration	User does Configuration
Create Cluster Resource Group API Create Cluster Resource Group Command	• Cluster Resource Services ensures the IP address does not exist on any node in the recovery domain. • Cluster Resource Services adds the IP address to every node in the recovery domain.	• Cluster Resource Services does not do anything with the IP address.
Delete Cluster API Delete Cluster Command	• The IP address is ended on the primary node if the cluster resource group is active. • Cluster Resource Services removes the IP address on all nodes in a cluster resource group's recovery domain.	• The IP address is ended on the primary node if the cluster resource group is active. • Cluster Resource Services does not do anything else with the IP address.
Delete Cluster Resource Group API Delete Cluster Resource Group Cluster Command	• Cluster Resource Services removes the IP address on all nodes in a cluster resource group's recovery domain.	• Cluster Resource Services does not do anything with the IP address.
Delete Cluster Resource Group CL command	• Cluster Resource Services removes the IP address on all nodes in a cluster resource group's recovery domain.	• Cluster Resource Services does not do anything with the IP address.
End Cluster Node API End Cluster Node Command	• If the cluster resource group is active and the node being ended is the primary node, Cluster Resource Services ends the IP address on the primary node after calling the exit program with the End Node action code. See the failover event for how other nodes in the recovery domain are handled.	• If the cluster resource group is active and the node being ended is the primary node, Cluster Resource Services ends the IP address on the primary node after calling the exit program with the End Node action code. See the failover event for how other nodes in the recovery domain are handled.
End Cluster Resource Group API End Cluster Resource Group Command	• Cluster Resource Services ends the IP address on the primary node after calling the exit program with the End action code.	• Cluster Resource Services ends the IP address on the primary node after calling the exit program with the End action code.
Failover event	• Cluster Resource Services starts the IP address on the new primary node before calling the exit program with the Start action code.	• Cluster Resource Services starts the IP address on the new primary node before calling the exit program with the Start action code.

API or cluster event	Cluster Resource Services does Configuration	User does Configuration
Initiate Switchover API Change Cluster Resource Group Primary Command	• Cluster Resource Services ends the IP address on the current primary node before calling the exit program with the Switchover action code. • Cluster Resource Services starts the IP address on the new primary node before calling the exit program with the Start action code.	• Cluster Resource Services ends the IP address on the current primary node before calling the exit program with the Switchover action code. • Cluster Resource Services starts the IP address on the new primary node before calling the exit program with the Start action code.
Node joining event	• If the cluster resource group is active and the node joining is a backup node, Cluster Resource Services ensures the IP address is not active on the joining node.	• If the cluster resource group is active and the node joining is a backup node, Cluster Resource Services ensures the IP address is not active on the joining node.
Partition merge event	• If the cluster resource group is active and the node(s) merging with the primary partition is a backup node, Cluster Resource Services ensures the IP address is not active on the merging node(s).	• If the cluster resource group is active and the node(s) merging with the primary partition is a backup node, Cluster Resource Services ensures the IP address is not active on the merging node(s).
Remove Cluster Node Entry API Remove Cluster Node Entry Command	• If the cluster resource group is active and the node being removed is the primary node, Cluster Resource Services ends the IP address on the primary node after calling the exit program with the Remove Node action code. See the failover event for how other nodes in the recovery domain are handled. • Cluster Resource Services removes the IP address on the node being removed.	• If the cluster resource group is active and the node being removed is the primary node, Cluster Resource Services ends the IP address on the primary node after calling the exit program with the Remove Node action code. See the failover event for how other nodes in the recovery domain are handled.
Remove Node from Recovery Domain API Remove Cluster Resource Group Node Entry Command	• Cluster Resource Services removes the IP address on the node being removed.	• Cluster Resource Services does not do anything with the IP address.
Start Cluster Resource Group API Start Cluster Resource Group Command	• Cluster Resource Services ensures the IP address exists on the primary node and all backup nodes. • Cluster Resource Services ensures the IP address is not active on any node » unless the allow active takeover IP address field is set to 1. « • Cluster Resource Services starts the IP address on the primary node before calling the exit program with the Start action code.	• Cluster Resource Services ensures the IP address exists on the primary node and all backup nodes. • Cluster Resource Services ensures the IP address is not active on any node in the recovery domain. • Cluster Resource Services starts the IP address on the primary node before calling the exit program with the Start action code.

When the Exit Program Ends

When an exit program is called with an action code, control can return to its caller because it set the success indicator and returned, had an unhandled exception, or the exit program job was cancelled.

Setting the Success Indicator and Returning

The returned value of the success indicator is used by the operating system in different ways depending upon the action code. For most action codes, anything other than **Successful** will result in the exit program being called again with an action code of Undo to backout the actions previously performed. There are two exceptions to this.

One, if an application exit program was called with an action code of Start, setting the success indicator to **Unsuccessful, attempt restart** will result in the exit program being called with Restart. Being called with an action code of Restart will occur as long as the restart count has not been reached. When the restart count is reached, failover occurs and the application is started on the first active backup node.

The exit program is not called with Restart if either an **Unsuccessful, do not attempt restart** indicator is returned, the exit program sets the success indicator to **Successful** and returns, or the cluster resource group is ended with the "End Cluster Resource Group (QcstEndClusterResourceGroup) API" on page 132.

Two, some action codes always proceed regardless of the exit program success indicator and the exit program is not called again with an action code of Undo. These are:

- Change Cluster Node Entry
- Delete
- Delete Command
- End Cluster Node
- Remove Node (only when removing a node from the cluster)
- Undo

If the exit program returns an unsuccessful indicator from Undo, the cluster resource group's status is set to Indoubt. » An exception to this case would be the unsuccessful switchover for a device cluster resource group.After switching back all the devices, if all of the devices are varied-on successfully on the original primary node, clustering calls the exit program on the original primary node with an action code of Start. If this call fails for any reason, an error message is sent to the cluster resource group's joblog but its status is not set to Indoubt. «

An Exception Occurs

An unhandled exception is treated the same way as an unsuccessful indicator. The exit program will be called again with either Restart or Undo except for the same action codes listed above where it is not called again with Undo.

If the exit program does not handle an exception while processing Undo, the cluster resource group's status is set to Indoubt.

Job is Cancelled

If the exit program job is cancelled and the exit program was performing the function of any action code other than Undo, Start, or Restart, it is treated as an unsuccessful indicator. The exit program is called with an Undo action code except for those action codes listed above where it is not called again with Undo.

If the exit program was cancelled while performing the function of Undo, the cluster resource group's status is set to Indoubt.

If the exit program was cancelled while performing the function of Start or Restart, the cluster resource group is ended; failover does not occur. It is the responsibility of the exit program cancel handler to also end any other jobs or subsystems it may have started.

An exit program job always has an associated cluster resource group job. It is the associated cluster resource group job that submits the exit program job. If the cluster resource group job is cancelled while an exit program is running, the exit program job is also cancelled. If the cluster resource group job is cancelled, the exit program is called with the End Node action code on the node where the job was cancelled.

Restarting an Application Cluster Resource Group Exit Program

Cluster Resource Services uses a restart count to control how often an active application will be restarted on the primary node before a failover occurs. The restart count is specified on the "Create Cluster Resource Group (QcstCreateClusterResourceGroup) API" on page 107 or "Change Cluster Resource Group (QcstChangeClusterResourceGroup) API" on page 86 APIs for application cluster resource groups. If the specified value is 0, the failed application will not be restarted on the primary node but failed over to the first backup. If the specified value is greater than 0, Cluster Resource Services will call the exit program with an action code of Restart after having initially called the exit program with an action code of Start. It will continue to do this for each failure, until the restart count has been reached. The exit program will be called with an action code of Restart if it returns from handling the Start action code in one of these ways:

- The exit program returns with the success indicator set to 2 (Unsuccessful, attempt restart).
- The exit program does not handle an exception.

Once the restart count has been reached, Failover will be attempted in order to start the application on the first active backup node. The restart count is reset only when the exit program is called with a Start action code. This occurs with the "Start Cluster Resource Group (QcstStartClusterResourceGroup) API" on page 164 or "Initiate Switchover (QcstInitiateSwitchOver) API" on page 135 or the failover event.

Multiple Action Codes

In most situations, cluster APIs or events result in the exit program being called with a single action code. When the exit program completes successfully, the exit program is not called again for that API or cluster event. There are several situations where successful completion results in the exit program being called twice. This occurs for active application cluster resource groups for the Initiate Switchover API and the failover cluster event. In both cases, the exit program is called on the new primary first with either the Switchover or Failover action code. During this time, the exit program should do any preparation work necessary to start the application but should not yet start the application. When the exit program returns with a successful indicator, it will be called a second time with the Start action code to start the application.

Another situation occurs when a cluster resource group is deleted using either the Delete Cluster Resource Group API or Delete Cluster Resource Group From Cluster command. The exit program will be called first with Verification Phase action code and then with the Delete action code. If the verification phase returns with a unsuccessful indicator, the exit program will not be called a second time and the cluster resource group will not be deleted.

Causes of the Failover Event

It is natural to think of the failover event being caused by the most obvious problem: a node fails. The node failure could be due to a hardware problem such as the loss of a processor or an environmental problem such as the loss of electrical power.

There are a wide variety of other things that can cause a failover event when it occurs on a node that is in a cluster resource group's recovery domain. For more information about failover events, see Clusters in the System Management topic.

- "Cluster APIs," on page 1 or Commands
  - End Cluster Node API
  - End Cluster Node Command

  - Remove Cluster Node Entry API when Cluster Resource Services is active on the node
  - Remove Cluster Node Entry Command
- iSeries operator actions when Cluster Resource Services is active
  - Pushing the IPL button on the front panel
  - Powering down the system (PWRDWNSYS)
  - Ending all subsystems (ENDSBS)
  - Ending the QSYSWRK subsystem (ENDSBS)
  - Change Cluster Recovery command
  - Ending the system (ENDSYS)
  - Ending TCP (ENDTCP)
  - Cancelling the QCSTCTL, QCSTCRGM, or a specific CRG job
  - Ending a TCP/IP interface used by clustering for communication with other nodes
- Failures
  - System hardware failure causing a machine check
  - System software failure causing a machine check
  - Communication line, router, and IOP failures for a communication line used by clustering for communication with other nodes
  - Environmental problems causing the system to fail (electrical power loss, hardware destruction by storms, earth quake, and so on)
  - An application exit program returns from handing the Start or Restart action code with the Success indicator set to **Unsuccessful, attempt restart** and the restart limit has been reached
  - An application exit program returns from handing the Start or Restart action code with the Success indicator set to **Unsuccessful, do not attempt restart**
  - An application exit program does not handle an exception while processing the Start or Restart action code and the restart limit has been reached

The failover event always calls the exit program so that the exit program is aware a member left the cluster. The exit program is called regardless of the state of the cluster resource group: active, inactive, or indoubt. Also, the exit program is called regardless of which member left the cluster: primary, backup, replicate » or peer «. The exit program must look at both the state of the cluster resource group and the role of the node that left in order to perform the correct action.

Cluster resource groups should failover in a particular order when a node failure occurs. That order is device cluster resource groups first, application resource groups, and then data cluster resource groups.

» Peer cluster resource groups failover in parallel with the other cluster resource group types. «

Partition Processing

A cluster enters a partition state when a failure occurs that cannot conclusively be identified as a node failure. Cluster Resource Services detects that communication with another node or nodes has been lost but cannot determine why. A classic example is the failure of a communication line between the systems.

The exit program is called when a cluster partitions. The membership status for each partitioned node in the recovery domain will be set to Partition. However, this is different for each cluster partition. For example, suppose we have a 2 node cluster with nodes A and B, both nodes are in a cluster resource group's recovery domain, and the cluster partitions. When the exit program on A is called, the recovery domain will indicate that A is active and B is partitioned. When the exit program on B is called, the recovery domain will indicate that B is active and A is partitioned.

» For primary-backup model cluster resource groups: «

- The action code seen by the exit program on each node depends upon whether the node is in the primary or secondary partition. The primary partition contains the cluster resource group's primary node. The secondary partition does not.
- All nodes in the primary partition of the cluster resource group's recovery domain will be passed an action code of Failover. All nodes in the secondary partition are passed an action code of End. Action code dependent data of Partition failure is passed in each case. These action codes are used whether the cluster resource group is active or inactive.

» For peer model cluster resource groups:
- All recovery domain nodes will be passed an action code of Failover. The access points that are active will remain active in all partitions.
- All configuration changes are not allowed when the recovery domain spans a network partition. «

Handling the Undo Action Code

When Cluster Resource Services processes an API or cluster event and an exit program is called, a failure either in the exit program or in Cluster Resource Services after the exit program ends results in an attempt to recover the prior state of the cluster resource group and its resilient resources.

Actions performed by Cluster Resource Services which changed the cluster resource group are backed out. The exit program is called with an action code of Undo so that actions it took can also be backed out.

If the exit program had nothing to do for an action code, its work to handle the Undo is trivial. Merely set the success indicator to Successful and return.

If the exit program has a failure and can back out its actions as part of handling the original action code, it may also have little or nothing to do when called with the Undo action code. Doing this back out as part of the original action code processing may be driven from the procedure which detected the problem, or from an exception handler, or from a cancel handler.

When the exit program handles the original action code successfully but Cluster Resource Services subsequently detects an error that requires the API or cluster event to be backed out, the Undo processing by the exit program becomes more involved. While the exit program is passed the action code it worked on before being called with Undo, there may be other information the exit program will have to obtain in order to successfully perform the back out. Any required back out information will have to be kept where a new job can be access it.

The format data passed to the exit program for Undo is exactly the same as was passed for the original action code except for the prior action code field.

A cluster resource group's status is returned to its original value if both the exit program and Cluster Resource Services handle the Undo action code successfully. If Cluster Resource Services is unable to back out changes or the exit program sets the success indicator to anything other than Successful, the status of the cluster resource group is set to Indoubt. When this occurs, someone such as an operator or programmer may have to be involved to determine what errors caused the problem.

Reasons an Exit Program Is Called

The table below shows the reasons an exit program is called and maps the reason to the **Action Code** parameter on the cluster resource group exit program. The third and fourth columns of the table give suggestions for the types of things a data or application cluster resource group exit program might do for an action code.

The following "Cluster Resource Group APIs" on page 61 or commands do not cause the exit program to be called:

- Change Cluster Resource Group API if the recovery domain is not changed
- Change Cluster Resource Group Command
- Distribute Information API
- List Cluster Resource Group Information API
- List Cluster Resource Groups API
- Display Cluster Resource Group Command

For a device cluster resource group, neither the replication provider nor the application provider need to supply an exit program. An exit program is optional. An exit program is required only if customer specific activities are required for resilient devices. Some examples of why a customer may wish to provide an exit program might include:

- When a cluster resource group is created or a node is added to the recovery domain, the exit program could perform configuration functions for devices not supported by the device cluster resource group.
- When a cluster resource group is started, the exit program could vary on devices not supported by the device cluster resource group.
- When a switchover or failover is done, the exit program could vary off devices on the current primary node for devices not supported by the device cluster resource group and vary them on for the new primary node.
- When a cluster resource group is deleted or a node is removed from the recovery domain, the exit program could delete configuration information previously created.
- Besides managing device configuration or varying devices on or off, the exit program could perform other functions that might be useful in synchronizing events between actions on a device and operator notification or application dependencies.

Table 2. Reasons an Exit Program Is Called

Reason an Exit Program Is Called	**Action Code Parameter Passed to Exit Program**	**Supplied by Replication Provider Exit Program Actions - Data » /Peer « Resilience**	**Supplied by Application Provider Exit Program Actions - Application Resilience**
Create Cluster Resource Group API or Create Cluster Resource Group Command This interface creates a cluster resource group object, which identifies a recovery domain.	1 (Initialize)	Put data on all nodes in the recovery domain. Prime all nodes in the recovery domain.	Put applications on all nodes in the recovery domain. Prime all nodes in the recovery domain.
Start Cluster Resource Group API or Start Cluster Resource Group Command This interface establishes resilience for a cluster resource group.	2 (Start)	• Start journaling. • Start replication. » If a Peer CRG is active, all peer nodes are actively replicating. «	• Start server jobs. • Keep track of server jobs started. This will be needed when server jobs are restarted or the End Cluster Resource Group API is called.
Application cluster resource group exit program ends abnormally or unexpectedly.	3 (Restart)	Not called.	• Restart server jobs if necessary.

Reason an Exit Program Is Called	Action Code Parameter Passed to Exit Program	Supplied by Replication Provider Exit Program Actions - Data >> /Peer << Resilience	Supplied by Application Provider Exit Program Actions - Application Resilience
End Cluster Resource Group API or End Cluster Resource Group Command This interface will disable resilience for a cluster resource group object. **Application ends** The Success indicator is sent to Successful and the application ends	4 (End)	• Stop replication. • End journaling.	• End server jobs. • End application resilience.
Delete Cluster Resource Group API or Delete Cluster Resource Group from Cluster Command	5 (Verification Phase)	Verify that the operation is ok to do.	Verify that the operation is ok to do.
	6 (Reserved)		
Delete Cluster Resource Group API or Delete Cluster Resource Group From Cluster Command This interface deletes a cluster resource group object from all nodes in the recovery domain. **Delete Cluster API or Delete Cluster Command** (if Cluster Resource Services is active) This interface deletes all cluster resource groups from all nodes. **Remove Cluster Node Entry API or Remove Cluster Node Entry Command** (if Cluster Resource Services is active and node being remove is not active in the cluster.) This interface deletes a cluster resource group object from all nodes in the recovery domain.	7 (Delete)	7 (Delete)	7 (Delete)

Reason an Exit Program Is Called	Action Code Parameter Passed to Exit Program	Supplied by Replication Provider Exit Program Actions - Data »/Peer« Resilience	Supplied by Application Provider Exit Program Actions - Application Resilience
Start Cluster Node API or Start Cluster Node Group Command This interface is used to start Cluster Resource Services on one or more nodes in the cluster. **Partition merge event** When a communication problem which caused a cluster to partition has been corrected, cluster partitions will merge back together.	8 (Rejoin)	• Resynchronize data • Start replication if the cluster resource group status is active (10). »For peer this should be done for all peer nodes.«	Start application if the cluster resource group status is Active (10)
Node failure or resource failure event **End Cluster Node API or End Cluster Node Command** The recovery domain node(s) which are not being ended see this action code. The node being ended sees the End Node action code. **Remove Cluster Node Entry API or Remove Cluster Node Entry Command** An active recovery domain node is being removed from the cluster.	9 (Failover)	»For Data:« • Get data objects to highest level of currency. • Redirect remote journal receivers. »For peer, this is just notification only.«	• Make sure exit program data contains all key information for application restart. This can be accomplished by the Change Cluster Resource Group API or Change Cluster Resource Group Command. • Use exit program data to restart application at last known point. Exit program data must contain enough key information for most effective restart. • Restart server jobs after data is current. Actually occurs when the Cluster Resource Services calls the cluster resource group exit program with an action code of 2 (Start) on the new primary only.
Initiate Switchover API or Change Cluster Resource Group Primary Command This API changes the current role of a node in the recovery domain of a cluster resource group by switching the access point from the primary node to the first backup.	10 (Switchover)	»For Data only:« • Stop replication on primary and journaling. • Continue replication on other active nodes in the recovery domain. This is a combination of 4 (End) and 9 (Failover). »Not valid for peer.«	• Make sure exit program data contains all key information for application restart before the Initiate Switchover API is called. • 10 (Switchover - Pre-exit program) – Stop server jobs. • 2 (Start - Post-exit program) – Use exit program data to restart application at last known point. Exit program data must contain enough key information for most effective restart. – Restart server jobs after data is current.

Reason an Exit Program Is Called	Action Code Parameter Passed to Exit Program	Supplied by Replication Provider Exit Program Actions - Data /Peer Resilience	Supplied by Application Provider Exit Program Actions - Application Resilience
Add Node to Recovery Domain API or Add Cluster Resource Group Node Entry Command This interface will add a node ID to the recovery domain of a cluster resource group.	11 (Add Node)	Exit program actions performed on the node being added: • If cluster resource group is Active, do 1 (Initialize) and 2 (Start) actions. A peer node will be an active access point. • If cluster resource group is Inactive, do 1 (Initialize) action.	Perform 1 (Initialize) on the node being added.
Remove Node from Recovery Domain API or Remove Cluster Resource Group Node Entry Command This interface will remove a node from the recovery domain of a cluster resource group. **Remove Cluster Node Entry API or Remove Cluster Node Entry Command** (will be seen on the node being removed if Cluster Resource Services is active on the node being removed) **Remove Cluster Node Entry API or Remove Cluster Node Entry Command** (will be seen on active cluster nodes if Cluster Resource Services is inactive on the node being removed and the API is run on an active node)	12 (Remove Node)	Exit program actions on the node being removed: • If the cluster resource group is Active and the node being removed is active in the cluster do 4 (End) and 7 (Delete). For peer this should be done on all active peer nodes. If the node being remove is not active in the cluster, this is just notification that the node is being removed from the recovery domain. • If cluster resource group is Inactive and the node being removed is active in the cluster , do 7 (Delete) action. If the node being remove is not active in the cluster, this is just notification that the node is being removed from the recovery domain.	Exit program actions on the node being removed: • If the cluster resource group is Active and the node being removed is active in the cluster, do 4 (End) and 7 (Delete). If the node being remove is not active in the cluster, this is just notification that the node is being removed from the recovery domain. • If cluster resource group is Inactive and the node being removed is active in the cluster, do 7 (Delete) action. If the node being remove is not active in the cluster, this is just notification that the node is being removed from the recovery domain.
Change Cluster Resource Group API or Change Cluster Resource Group Command This interface changes some of the attributes of a cluster resource group. Only if the recovery domain is changed will the cluster resource group exit program be called.	13 (Change)	• Redirect replication if necessary. • Redirect journaling if necessary. For peer if the cluster resource group is Active (10) and the node is being changed from replicate to peer perform a 2 (Start) on the new peer node. If the node is being changed from peer to replicate perform an 4 (End) on the new replicate node.	

Reason an Exit Program Is Called	Action Code Parameter Passed to Exit Program	Supplied by Replication Provider Exit Program Actions - Data /Peer Resilience	Supplied by Application Provider Exit Program Actions - Application Resilience
Delete Cluster Resource Group CL command This command deletes a cluster resource group object from the node running the command. This is not a distributed request. **Delete Cluster API or Delete Cluster Command** (if Cluster Resource Services is inactive) **Remove Cluster Node Entry API or Remove Cluster Node Entry Command** (if Cluster Resource Services is inactive on the node being removed and the API is run on that node)	14 (Delete Command)		
	15(Undo)	Rollback operations from previous request.	Rollback operations from previous request.
End Cluster Node API or End Cluster Node Command is used to end Cluster Resource Services on a node in the recovery domain. **Job cancelled** A Cluster Resource Services job is cancelled.	16(End Node)	On the node being ended: • Do End (4) and Change (13) if the cluster resource group status is Active (10). • Do Change (13) if the cluster resource group status is Inactive(20) or Indoubt (30). For data cluster resource group if the node assigned the primary role is ended, exit program actions on the first backup: • If the cluster resource group status is Active (10) do Failover (9). • If the cluster resource group status is Inactive (20) or Indoubt (30) do Change (13).	On the node being ended: • Do End (4) and Change (13) if the cluster resource group status is Active (10). • Do Change (13) if the cluster resource group status is Inactive (20) or Indoubt (30). If the node assigned the primary role is ended, exit program actions on the first backup: • If the cluster resource group status is Active (10) do Failover (9). • If the cluster resource group status is Inactive (20) or Indoubt (30) do Change (13).
Add Cluster Resource Group Device Entry API or Add Cluster Resource Group Device Entry Command A resilient device entry is added to a cluster resource group	17(Add Device Entry)	Does not apply to a data or peer cluster resource group.	Does not apply to an application cluster resource group

Reason an Exit Program Is Called	Action Code Parameter Passed to Exit Program	Supplied by Replication Provider Exit Program Actions - Data » /Peer « Resilience	Supplied by Application Provider Exit Program Actions - Application Resilience
Remove Cluster Resource Group Device Entry API or Remove Cluster Resource Group Device Entry Command A resilient device entry is removed from a cluster resource group	18(Remove Device Entry)	Does not apply to a data » or peer « cluster resource group	Does not apply to an application cluster resource group
Change Cluster Resource Group Device Entry API or Change Cluster Resource Group Device Entry Command Information for a resilient device entry is being changed	19(Change Device Entry)	Does not apply to a data » or peer « cluster resource group	Does not apply to an application cluster resource group
Change Cluster Node Entry API or Change Cluster Node Entry Command The status of a node is being changed.	20(Change Node Status)	» For data cluster resource group « if the primary had failed and its status is being changed, start the cluster resource group.	If the primary had failed and its status is being changed, start the cluster resource group.
Primary node failure or resource failure event Failover is cancelled by failover message queue **End Cluster Node API or End Cluster Node Command** Primary node is ended and failover cancelled by failover message queue. The recovery domain node(s) which are not being ended see this action code. The node being ended sees the End Node action code.	21 (Failover Cancelled)	» Not applicable to peer cluster resource group. « • Stop replication. • End journaling.	• End server jobs. • End application resilience.
Primary node failure Failover is cancelled by failover message queue **End Cluster Node API or End Cluster Node Command** Primary node is ended and failover cancelled by failover message queue. The recovery domain node(s) which are not being ended see this action code. The node being ended sees the End Node action code.			

Action Code Cross Reference

Some action codes are used by more than one API or cluster event. The following table is a cross reference between an action code and which API or cluster event uses it. The action code dependent data value is listed in parenthesis after each API and cluster event. Those with no specified dependent data value have a value of No Information (0).

Table 3. API and Cluster Event to Action Code Cross Reference

Action Code	API, Command, or Cluster Event that Uses the Action Code	Cluster Resource Group Type the Action Code Applies to
1 - Initialize	• Create Cluster Resource Group API • Create Cluster Resource Group Command	All cluster resource group types
2 - Start	• Start Cluster Resource Group API • Start Cluster Resource Group Command • The second action code on the new primary for Initiate Switchover API for an active application cluster resource group • The second action code on the new primary for Failover event for an active application cluster resource group • » The third action code on the original primary for a unsuccessful Switchover event for an active device cluster resource group. After switching back all the devices, if all of the devices are varied-on successfully on the original primary node, clustering calls the exit program on the original primary node with an Start action code. If the failure occurs before the first call to the exit program with action code of switchover or failover, the user will not see the exit program call with Undo, but clustering will still make the call on the original primary with action code Start after successful vary-on. «	All cluster resource group types
3 - Restart	Exit program failure when processing the Start action code	Application cluster resource group
4 - End	• End Cluster Resource Group API (0 - No information) • End Cluster Resource Group Command • » For primary-backup model cluster resource groups only. « Cluster partition event for the nodes in the secondary partition for both active and inactive cluster resource groups (3 - Partition failure) • When an application cluster resource group exit program ends with a Success return code while processing the Start or Restart action codes (9 - Resource end)	All cluster resource group types
5 - Verification Phase	• Delete Cluster Resource Group from Cluster Command • Delete Cluster Resource Group API	All cluster resource group types

Action Code	API, Command, or Cluster Event that Uses the Action Code	Cluster Resource Group Type the Action Code Applies to
7 - Delete	• Delete Cluster Resource Group API (0 - No information) • Delete Cluster API (10 - Delete cluster) • Delete Cluster Resource Group From Cluster Command • Delete Cluster Command • For primary-backup model cluster resource groups only. Remove Cluster Node Entry API or Remove Cluster Node Entry Command (15 - Remove passive node) When a node being removed from the cluster is not active and the node is the only recovery domain node or it is the primary and there are no backups defined.	All cluster resource group types
8 - Rejoin	• Cluster partitions are merging (1 - Merge) • A node that was ended or failed is started (2 - Join)	All cluster resource group types
9 - Failover	See "Causes of the Failover Event" on page 206 for a list of things that cause the failover event (4 - Node failure, 5 - Member failure, 6 - End node, 7 - Remove node, 8 - Application failover)	All cluster resource group types
10 - Switchover	• Initiate Switchover API • Change Cluster Resource Group Primary Command	All cluster resource group types except peer
11 - Add Node	• Add Node to Recovery Domain API • Add Cluster Resource Group Node Entry Command	All cluster resource group types
12 - Remove Node	• Remove Cluster Node Entry API or Remove Cluster Node Entry Command All nodes see this action code if the node being removed is inactive. The action code dependent data will be 15 - Remove passive node. If removed node is active, the node being removed sees this action code while the other nodes see the Failover action code (7 - QcstRemoveNode). • Remove Node From Recovery Domain API Remove Cluster ReSource Group Node Entry Entry Command (11 - Remove recovery domain node)	All cluster resource group types
13 - Change	• Change Cluster Resource Group API • Change Cluster Resource Group Command	All cluster resource group types
14 - Delete Command	• Delete Cluster Resource Group command • Delete Cluster API (when Cluster Resource Services is inactive) • Remove Cluster Node Entry API used on a node where Cluster Resource Services is not running	All cluster resource group types

Action Code	API, Command, or Cluster Event that Uses the Action Code	Cluster Resource Group Type the Action Code Applies to
15 - Undo	An Undo action code is used whenever the exit program is ended due to an unhandled exception or returns with a non successful return code except for these action codes which never call the exit program a second time with Undo: • 7 - Delete • 12 - Remove Node (if the node being removed from the cluster) • 14 - Delete Command • 15 - Undo • 16 - End Node • 20 - Change Cluster Node Entry • 22 - Cancel	All cluster resource group types.
16 - End Node	• End Cluster Node API (0 - No information) • End Cluster Node Command • A cluster resource group job is cancelled on a node (5 - Member failure)	All cluster resource group types
17 - Add Device Entry	• Add Cluster Resource Group Device Entry API • Add Cluster Resource Group Device Entry Command	Device cluster resource group
18 - Remove Device Entry	• Remove Cluster Resource Group Device Entry API • Remove Cluster Resource Group Device Entry Command	Device cluster resource group
19 - Change Device Entry	• Change Cluster Resource Group Device Entry API • Change Cluster Resource Group Device Entry Command	Device cluster resource group
20 - Change Node Status	• Change Cluster Node Entry API • Change Cluster Node Entry Command when used to change the status of a cluster node to Failed.	All cluster resource group types
21 - Failover Cancelled	The primary node failed and the failover was cancelled through the use of the failover message queue. See “Causes of the Failover Event” on page 206 for a list of things that cause the failover event (4 - Node failure, 5 - Member failure, 6 - End node, 7 - Remove node, 8 - Application failover)	All cluster resource group types » except peer «
22 - Cancel	» N/A «	» N/A «

Exit program introduced: V4R4

Top | “Cluster APIs,” on page 1 | APIs by category

Concepts

These are the concepts for this category.

Cluster APIs—Introduction

An iSeries(TM) **cluster** is defined as a collection of complete systems that work together to provide a single, unified computing capability. A cluster is identified by a 10-character name. The cluster is comprised of one or more **cluster nodes**. A cluster node is an iSeries system that is a member of a cluster. Each cluster node is identified by an 8-character cluster node identifier that is associated with a set of IP addresses representing an iSeries system.

Cluster communications running over IP provides the communications path between cluster services on each node in the cluster. Cluster Resource Services requires that the loopback IP address on each node be active. The set of cluster nodes that have been configured for the cluster is referred to as the **cluster membership list**.

Whenever communication with a node is lost but node or cluster resource services failure cannot be guaranteed, a cluster becomes **partitioned**. A cluster may be separated into multiple partitions. While partitioned, some cluster operations may be restricted.

High availability through clustering is accomplished through the implementation of **resilient resources**. A resilient resource is any system resource supported by clustering that is available on more than one node in the cluster. If a node in the cluster that is the primary access point for a particular set of resilient resources should fail, another node that is defined to be the backup for that set of resources will become the access point. The definition of the relationship between the nodes associated with a set of resilient resources is found in the cluster resource group (CRG) object. Cluster resource groups are distributed and coordinated across the nodes in the cluster. Cluster resource groups contain a “Recovery Domain” on page 62.

» **Cluster Administrative Domain** will allow a cluster administrator to maintain a consistent operational environment across the cluster or a subset of the cluster. «

Example Source Code

Example control language command source has been provided in the base operating system option 7 (Example Tools Library, QUSRTOOL). These commands interface to the Cluster Resource Services APIs. See member, TCSTINFO, in file QUSRTOOL/QATTINFO for more information. » i5/OS(R) supported CL commands can be found by doing ″GO CMDCLU″ from the command entry screen. «

In addition, example application cluster resource group exit program source code can be found in the TCSTAPPEXT and TCSTDTAARA members in file QUSRTOOL/QATTSYSC.

Sample program source has been included which will create the QCSTHAAPPI, QCSTHAAPPO data areas and the QACSTOSDS object specifier file. See member TCSTDTAEXT in QUSRTOOL/QATTSYSC for the source. This is not a complete program as written. Data specific to the application must be added since each application has different resiliency requirements. » See High Availability and Clusters  for information on making applications highly available and cluster-proven applications. This website also contains more information on QCSTHAAPI data area, QCSTHAAPPO data area, and QACSTOSDS object specifier file. «

Terminology

Access point. » For primary-backup model, « the primary point of access for a resilient resource. If that resource fails, one of the backup resources will become the primary access point. » For peer model, the point of access for a resource. All nodes of type peer, are a point of access for the resource. «

» Administrative domain. Monitors and synchronizes changes to selected resources within a cluster. Cluster administrative domain provides easier management and synchronization of attributes for resources that are shared within a cluster, such as environment variable or user profiles.

Application takeover IP address. A floating address that is to be associated with an application.

Clustered hash table. Non-persistent data that can be shared and replicated between cluster nodes using the "Clustered Hash Table APIs" on page 169.

Cluster resource group exit program. A program which handles action codes that are passed to it.

Cluster resource. A resource that is available on more than one cluster node.

Cluster resource group. A grouping of cluster resources. The group describes a recovery domain and supplies the name of the cluster resource group exit program that manages the movement of an access point.

Device domain. A subset of nodes in a cluster grouped together to share physical hardware resources or the logical resources associated with the physical hardware.

Partition. Happens when you lose contact between one or more nodes in the cluster and a failure of lost nodes cannot be confirmed.

Peer model. Cluster resource groups of this model define nodes in the recovery domain with a specific role of either peer or replicate. The peer nodes are available to be the access point for the cluster resource groupbut .

Primary-backup model. Cluster resource groups of this model define nodes in the recovery domain with a specific role of either primary, backup or replicate. The primary and backup nodes are available to be the access point for the cluster resource group.

Recovery domain. A subset of nodes in a cluster grouped together to provide availability for one or more resources. A domain represents the nodes of the cluster where a cluster resource exists. See "Cluster Resource Group APIs" on page 61 for more information.

Resilient resource. Resources that are recoverable by Cluster Resource Services. Three types of system resources that can be resilient are (1) objects being replicated between nodes, (2) applications using a takeover IP address, which can be switched from one node to another, and (3) hardware devices which can be switched from one node to another.

Server takeover IP address. A takeover IP address for servers associated with the relational database name in the device description for an auxiliary storage pool.

Singleton cluster. A one node cluster or a cluster with only one active node.

Top | "Cluster APIs," on page 1 | APIs by category

Cluster Resource Services Characteristics

A set of APIs is provided to enable a developer to produce cluster management facilities and to integrate the facilities with other system management capabilities. The APIs provide access to the i5/OS(R) Cluster Resource Services functions and are described in the sections that follow the introduction.

CL commands for clustering are available in the operating system, should you like to experiment with clustering before using the Cluster APIs.

The Cluster Resource Services components are:

1. Cluster Control (CCTL)

CCTL provides configuration, activation, and management functions for the cluster and nodes in the cluster through a set of APIs. These APIs are described in “Cluster Control APIs” on page 1.

2. Cluster Resource Group Manager (CRGM)

   CRGM provides object management functions for the cluster resource group (*CRG) object through a set of APIs. These APIs are described in “Cluster Resource Group APIs” on page 61.

3. Cluster Resource Group Exit Program

   This user provided program is specified on the definition of a cluster resource group. The program is responsible for handling all of the action codes that are passed to it by the Cluster Resource Group Manager. The action codes are described in the APIs that result in a call to the exit program. More detail about exit programs can be found in “Cluster Resource Group Exit Program” on page 185.

4. Clustered Hash Table

   The clustered hash table server enables sharing and replicating non-persistent data between cluster nodes using the Clustered Hash Table APIs. These APIs are described in “Clustered Hash Table APIs” on page 169.

5. » Cluster Administrative Domain

   The cluster administrative domain will allow a consistent operational environment across the cluster or a subset of the cluster through the use of commands and Integrated Operating Environments APIs. These APIs are described in Integrated Operating Environments APIs.«

Nodes in a cluster can have different releases of the operating system installed on them. The function from a newer version of the operating system cannot always be used on nodes that have an older version of the operating system. A cluster version manages this. More information can be found in “Cluster Version” on page 224.

Top | “Cluster APIs,” on page 1 | APIs by category

Cluster Resource Services Job Structure

Cluster Resource Services consists of a set of multi-threaded jobs. When clustering is active on an iSeries,(TM) the jobs will be run in the QSYSWRK subsystem. The jobs run using the » QDFTSVR « job description. All Cluster Resource Services jobs automatically provide a joblog to aid in problem determination. For these jobs, the LOG parameter of the job description is overridden. An exit program job » may or may not « produce a joblog. To provide a joblog, change the LOG parameter of the job description to a level that will produce a joblog.

1. Cluster Control consists of one job. The job is named QCSTCTL and runs under the QSYS user profile.
2. Cluster Resource Group Manager consists of one job. The job is named QCSTCRGM and runs under the QSYS user profile.
3. Cluster Resource Groups consists of one job per cluster resource group object. The job name is the same as the cluster resource group name and runs under the QSYS user profile.

When using clustering, the multithreaded job action (QMLTTHDACN) system value must be set to either 1 or 2. See Display System Value (DSPSYSVAL) command for more information.

Most cluster resource group APIs result in a separate job being submitted using the user profile specified when the cluster resource group was created. The exit program defined in the cluster resource group is executed in the submitted job. » The job is submitted to the job queue defined in the job description specified in the user profile for the cluster resource group. « By default, the jobs are submitted to the QBATCH JOBQ. In general, this job queue is used for production batch jobs and will delay or prevent completion of the exit programs. In order to allow the APIs to run effectively, create a separate user profile, job description, and job queue for use by cluster resource groups. Specify the new user profile for all cluster resource groups created. The same program is executed on all nodes within the recovery domain defined for the cluster resource group using the distributed activity group support provided by Cluster Resource Services.

One or more batch jobs are also submitted for a device cluster resource group if devices must be varied off or on. Varying a device off or on can occur as a result of a failover event or because the Initiate Switchover API was called. In the case of the "Initiate Switchover (QcstInitiateSwitchOver) API" on page 135, the batch job runs under the same user profile as the one that called the API. » During a failover event this job runs under the QSYS user profile. To determine the subsystem this job runs under look at the job description associated with the QSYS user profile. «

» To reactivate a cluster resource group job with a cluster version of 4 or less required clustering to be ended and restarted on the node. Cluster version 5 allows a cluster user to activate the cluster resource group job through the Change Cluster Recovery (CHGCLURCY) command with ACTION(*STRCRGJOB) provided clustering is still active on the node. A CRG job can now be reactivated without ending and starting the cluster on the node. «

Behavior of Cluster Resource Services APIs

Most Cluster Resource Services APIs have an asynchronous behavior. In general, whenever a user program calls a Cluster Resource Services API, the API will:

- Check the status of the Cluster Resource Services that will be used to process the API request. If Cluster Resource Services is not active, the API will fail. See the specific API to determine if this condition applies.
- Validate the API parameters and check authorities.
- Validate that a restricted API is not being called from a cluster resource group exit program (see the description of each API to see if this applies).
- Assign a handle to the request. The handle is defined by the **Request Handle** parameter on the Cluster Resource Services APIs.
- Send the request handle and request to the Cluster Resource Services for processing.
- Return to the API caller with the request handle.
- Process the request on the caller's behalf. When the processing is complete, one or more messages are sent to the user queue specified by the **Results Information** parameter. Messages indicate whether the processing was successful or unsuccessful. Each message that Cluster Resource Services sends has a specific format. See "Cluster APIs Use of User Queues" for more information on this format. Part of the message key will be the Request Handle which is returned to the API user.
- Run the API on the originating node, one or more remote nodes, or both.
- Respond to the API request by returning messages to the requestor's user queue on the originating node.

If the node originating a request fails, the request will not be handled on any nodes in the cluster.

Top | "Cluster APIs," on page 1 | APIs by category

Cluster APIs Use of User Queues

Functions performed by APIs with a parameter called Results Information operate asynchronously and will have data sent to a user queue once the API has finished processing. The user queue must be created before calling the API. User queues are created with the Create User Queue (QUSCRTUQ) API. The queue cannot be in an independent auxiliary storage pool. The queue must be created as a KEYED queue. The key for the user queue is described in the format of the user queue entry. The user queue name is passed to the API. The user queue may be in either system or user domain storage. Suggested values for each parameter is shown in the comments.

```
          Parm:               Value of Parm:
QUSCRTUQ (
          UserQueueName,
          ExtendedAttr,
          QueueType,        /*    K            Keyed */
```

```
KeyLength,        /*    28                */
MaxMsgSize,       /*    64000             */
InitialNumMsg,    /*    1                 */
AddtNumMsg,       /*    1                 */
PublicAuth,       /*    *EXCLUDE          */
TextDescription,
Replace,          /*    *NO               */
ErrorCode,
Domain,           /*    *USER             */
Pointers,         /*    *NO               */
NumberQueueExt,   /*    -1                */
ReclaimStg        /*    0                 */
);
```

User queue key format

The following is the format of the user queue key. All user queue entries have the same format for the Cluster APIs that support the Results Information parameter.

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	CHAR(10)	Entry type
10	A	CHAR(2)	Entry identifier
12	C	CHAR(16)	Request handle

User queue entry format

The following is the format of the user queue entry. All user queue entries have the same format for the Cluster APIs that support the Results Information parameter.

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Message format version ID
4	4	BINARY(4)	Message type
8	8	CHAR(30)	API name
38	26	CHAR(7)	Message identifier
45	2D	CHAR(35)	Reserved
80	50	CHAR(8)	Failing node ID
88	58	CHAR(10)	External object name
98	62	CHAR(2)	Reserved
100	64	BINARY(4)	Offset to message data
104	68	BINARY(4)	Message data length
108	6C	CHAR(*)	Message data

Field Descriptions

API name. The name of the API that caused the results to be sent to the user queue.

Entry identifier. Format of the data. This is set by the i5/OS(R). Valid values are:

00 Data distributed by the cluster APIs.

Entry type. The entry type on the user queue. This value is set by the i5/OS. Valid values are:

**CRS* Cluster Resource Services sent the results information to the user queue.

External object name. The name of the object that was not successfully processed.

Failing node ID. The cluster node that detected the condition being reported.

Message data. The message data for the message identifier. The format of this field depends on the message identifier. The Retrieve Message (QMHRTVM) API can be used to determine the format of the message data for each message identifier.

Message data length. The length of the message data for the message identifier.

Message format version ID. The version of the message format that is being used.

Message identifier. An i5/OS message identifier associated with a message description defined by i5/OS.

Message type. The type of message sent to the user queue. Any diagnostic or information messages returned as a result of an API will appear before the completion message. The valid values are:

1 Diagnostic
2 Information
3 Completion
4 User data

Offset to message data. The number of bytes from the start of the user queue entry to the data.

Reserved. Reserved for future use. Set to hexadecimal zeroes.

Request handle. A unique identifier assigned by Cluster Resource Services and returned to the caller of the API.

Top | "Cluster APIs," on page 1 | APIs by category

Using Results Information

Messages placed on the user queue should be handled by user written programs. The messages will indicate the results of the function requested by the API. There are three classes of completion results:

- Successful - The request completed successfully on all nodes that performed the function.
- Partially Successful - The request did not complete successfully on all of the nodes that performed the function. In some instances, partial success is caused by one or more nodes in an inactive state. The user program needs to determine if a partially successful operation is considered successful or unsuccessful.
- Unsuccessful - The request failed on all nodes that performed the function.

Results that are partially successful or unsuccessful will have diagnostic or informational messages describing the causes of failure and the node that failed to process the request. See the description of each API for the possible messages returned to the user queue.

General Information Applicable to Cluster APIs

The following parameters on Cluster APIs must be valid object names and uppercase only:

- Node ID
- Cluster name
- Cluster resource group name
- Job name specified for an application cluster resource group
- Device domain name
- Site name

To qualify as a valid object name, the first character must be alphabetic (A-Z) or one of the special characters, $, @, or #. The remaining characters are the same as the first character, but can also include 0 through 9, underscores and periods.

Top | "Cluster APIs," on page 1 | APIs by category

Cluster Version

Terminology for Cluster Version

Cluster version. The cluster version identifies the communication level of the nodes in the cluster, the ability of a node to join the cluster and the ability of the cluster to support new function. It is composed of two parts, the **version** and the **modification level**. There are two representations of the cluster version, Current cluster version and Potential node version.

Current cluster version. The version at which the nodes in the cluster are actively communicating with each other. This value in conjunction with the Potential node version determines which nodes can join in the cluster. This value also determines the cluster's ability to use new functions supported by the node's potential node version. It is set when the cluster is created and can be changed by the "Adjust Cluster Version (QcstAdjustClusterVersion) API" on page 11.

Modification level. The modification level further identifies the version at which the nodes in the cluster communicate. It is updated when code changes that impact the communication between the cluster nodes are applied.

Potential node version. The version at which the node is capable of communicating with the other nodes in the cluster. This is the value associated with the cluster code installed on the node. It will be used to determine if the node can join a cluster. If communications have not yet been established with the node (status of New), then the potential node version will be reported as 0.

Version Restrictions

A cluster can be composed of cluster nodes that are at different cluster versions. There are, however, restrictions that are enforced:

1. **New cluster version function is not available until all cluster nodes are at the new cluster version.** The current cluster version must be equal to the new functions version. This implies the potential node version of all the nodes must be at the same version. For example, assume new function is provided in cluster version 2. The new function of cluster version 2 cannot be used if the current cluster version equals 1. The current cluster version cannot be adjusted to the new level until all cluster nodes are upgraded to version 2.
2. **If N is the current cluster version, only nodes with potential node version of N and N+1 can join the cluster.** N is defined when the cluster is created. This will either be the potential node version of the node that originated the create cluster, or the potential node version previous to the node that

originated the create cluster request. For example if it is desired to have cluster nodes with a previous potential node version join the cluster, one of the following must be done:

- Create the cluster on the node with the previous potential node version and add nodes with a higher potential node version to the cluster. The potential node version of the node being added must only be one version higher.
- Create the cluster on the node with the higher potential node version specifying a target cluster version of -1. Then add the nodes with a lower potential node version to the cluster. The nodes being added must only be one version difference.

3. **A cluster will run protocols at the lowest potential node version (N) only.** This is defined when the cluster is first created. N can either be set to the potential node version running on the node that originated the create request or one potential node version previous to the originator's potential node version.

» When adjusting the cluster version, the version is adjusted to be one level greater than the current cluster version. All nodes must have the same potential cluster version before the cluster version can be adjusted. «

Relationship of cluster version to i5/OS VRMs

Cluster Version	i5/OS VRM
» 5	V5R4M0 «
4	V5R3M0
3	V5R2M0
2	V5R1M0

How to view the cluster versions

The current cluster version and the potential node version are retrieveable through the List Cluster Information and the Retrieve Cluster Information APIs.

Summary of API changes by cluster version

This documents the actual changes to the parameters on the APIs. The intent is to show which changes are allowed for the value of the current cluster version (CCV). The API changes for CCV 3 are only allowed on a V5R2M0 operating system or greater.

API changes allowed for CCV 5 (implies all of the below plus the following):

API name	Brief Description
CRG APIs	
"Cluster Resource Group APIs" on page 61	• Support new Peer cluster resource group.
"Create Cluster Resource Group (QcstCreateClusterResourceGroup) API" on page 107	• Update RGDC0200 format to add allow active takeover IP address.
"List Cluster Resource Group Information (QcstListClusterResourceGroupIn) API" on page 141	• Update LRGI0200 format to return allow active takeover IP address.

API name	Brief Description
CRG APIs	
"Cluster Resource Group Exit Program" on page 185	• Update EXTP0100 format to include allow active takeover IP address.«

API changes allowed for CCV 4 (implies all of the below plus the following):

API name	Brief Description
CRG APIs	
"Change Cluster Resource Group (QcstChangeClusterResourceGroup) API" on page 86	• Update RGDC0300 format to allow changes for site name and data port IP addresses when changing current recovery domain. • Allow change to the new EXTP0200 format for cluster resource group exit program. • Support a new value (-2 = no change) for the node role field.
"Create Cluster Resource Group (QcstCreateClusterResourceGroup) API" on page 107	• Update RGDC0300 format to add site name and data port IP addresses to the recovery domain array. • Add a new EXTP0200 format for cluster resource group exit program.
"List Cluster Resource Group Information (QcstListClusterResourceGroupIn) API" on page 141	• Add a new LRGI0300 format to return site name and data port IP addresses for the recovery domain array.
"Cluster Resource Group Exit Program" on page 185	• Support a new EXTP0200 format to include site name and data port IP addresses for the recovery domain array.

API changes allowed for CCV 3 (implies all of the below plus the following):

API name	Brief Description
CCTL APIs	
All APIS that support the Request Information Queue (RIQ) parameter	The RIQ is not allowed to be in an independent auxiliary storage pool.
"Start Cluster Node (QcstStartClusterNode) API" on page 57	If any node that had been previously but not currently ACTIVE starts itself, it will be able to rejoin the current active cluster if it can find other active node in the cluster, otherwise it will become a singleton cluster.
CRG APIs	
Following objects: • Request Information Queue (RIQ) • Cluster Resource Group Exit Program • Distribute Information Queue	The identified objects are not allowed to be in an independent auxiliary storage pool.

API name	Brief Description
CCTL APIs	
"Add Cluster Resource Group Device Entry (QcstAddClusterResourceGroupDev) API" on page 75	• Primary and secondary auxiliary storage pools can be specified in a resilient device cluster resource group. • A new value is added to the configuration object online field. • If the auxiliary storage pool has been created, the configuration object online value is validated. • If a server takeover IP address is specified, it must exist on all nodes in the recovery domain if the cluster resource group is active. • If a data base name is specified in the configuration object, it must be the same on all nodes in the recovery domain.
"Add Node To Recovery Domain (QcstAddNodeToRcvyDomain) API" on page 81	• If the node being added is the new primary node, all auxiliary storage group members must be specified as device entries in a resilient device cluster resource group. • If a server takeover IP address is specified, it must exist on all nodes in the recovery domain if the cluster resource group is active. • If a data base name is specified in the configuration object, it must be the same on all nodes in the recovery domain.
"Change Cluster Resource Group (QcstChangeClusterResourceGroup) API" on page 86	• If the primary node is being changed in the current recovery domain, all auxiliary storage group members must be specified as device entries in a resilient device cluster resource group. • New RGDC0110 format in support of failover message queue.
"Change Cluster Resource Group Device Entry (QcstChgClusterResourceGroupDev) API" on page 102	• A new value is added to the configuration object online field. • If the auxiliary storage pool has been created, the configuration object online value is validated. • If a server takeover IP address is specified, it must exist on all nodes in the recovery domain if the cluster resource group is active.
"Create Cluster Resource Group (QcstCreateClusterResourceGroup) API" on page 107	• Primary and secondary auxiliary storage pools can be specified in a resilient device cluster resource group. • A new value is added to the configuration object online field. • If the auxiliary storage pool has been created, the configuration object online value is validated. • If a data base name is specified in a configuration object, it must be the same on all nodes in the recovery domain. • The server takeover IP address must be unique. It can only be associated with a Primary auxiliary storage pool.
"Initiate Switchover (QcstInitiateSwitchOver) API" on page 135	• All auxiliary storage group members must be specified as device entries in a resilient device cluster resource group.
"List Cluster Resource Groups (QcstListClusterResourceGroups) API" on page 152	• Information about auxiliary storage pool device types is returned. • Information about failover message queue is returned. • Information about server takeover IP address is returned. • The user space cannot be in an independent auxiliary storage pool.
"Remove Cluster Resource Group Device Entry (QcstRmvClusterResourceGroupDev) API" on page 156	If the cluster resource group is active, all members of an auxiliary storage pool group must be removed at the same time.

API name	Brief Description
CCTL APIs	
"Remove Node From Recovery Domain (QcstRemoveNodeFromRcvyDomain) API" on page 160	• If the node being removed is the primary node, all auxiliary storage group members must be specified as device entries in a resilient device cluster resource group.
"Start Cluster Resource Group (QcstStartClusterResourceGroup) API" on page 164	• All auxiliary storage group members must be specified as device entries in a resilient device cluster resource group. • The value of the configuration object online field must be correct for the type of auxiliary storage pool. • If a data base name has been specified for a configuration object, it must be the same on all active nodes in the recovery domain. • If a server takeover IP address is specified, it must exist on all nodes in the recovery domain.
"Cluster Resource Group Exit Program" on page 185	• Format changes • New action code - Verification Phase • New action code - Failover Cancelled
Clustered Hash Table APIs	
"Connect Clustered Hash Table (QcstConnectCHT) API" on page 171	New API
"Disconnect Clustered Hash Table (QcstDisconnectCHT) API" on page 172	New API
"Generate Clustered Hash Table Key (QcstGenerateCHTKey) API" on page 173	New API
"List Clustered Hash Table Keys (QcstListCHTKeys) API" on page 174	New API
"Retrieve Clustered Hash Table Entry (QcstRetrieveCHTEntry) API" on page 179	New API
"Store Clustered Hash Table Entry (QcstStoreCHTEntry) API" on page 182	New API

API changes allowed for CCV 2:

API name	Brief Description
CCTL APIs	
"Add Device Domain Entry (QcstAddDeviceDomainEntry) API" on page 8	New API
"Change Cluster Resource Services (QcstChgClusterResourceServices) API" on page 19	New API
"Change Cluster Node Entry (QcstChangeClusterNodeEntry) API" on page 14	New format STSC0100 for changing node status from Partition to Failed.
"List Device Domain Information (QcstListDeviceDomainInfo) API" on page 41	New API

API name	Brief Description
CCTL APIs	
"Retrieve Cluster Information (QcstRetrieveClusterInfo) API" on page 50	New API
"Remove Device Domain Entry (QcstRemoveDeviceDomainEntry) API" on page 47	New API
CRG APIs	
"Add Cluster Resource Group Device Entry (QcstAddClusterResourceGroupDev) API" on page 75	New API
"Change Cluster Resource Group (QcstChangeClusterResourceGroup) API" on page 86	New RGDC0300 format in support of device cluster resource groups
"Change Cluster Resource Group Device Entry (QcstChgClusterResourceGroupDev) API" on page 102	New API
"Create Cluster Resource Group (QcstCreateClusterResourceGroup) API" on page 107	• New format RGDI0300 to support device cluster resource groups • New configure takeover IP address option • New distribute information queue field
"Distribute Information (QcstDistributeInformation) API" on page 127	New API
"List Cluster Resource Group Information (QcstListClusterResourceGroupIn) API" on page 141	New format LRGI0200 for device cluster resource groups
"Remove Cluster Resource Group Device Entry (QcstRmvClusterResourceGroupDev) API" on page 156	New API

Top | "Cluster APIs," on page 1 | APIs by category

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:

```
IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan
```

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

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

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

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

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

© Copyright IBM Corp. 1998, 2006

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 information 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.

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.

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

(C) IBM 2006. Portions of this code are derived from IBM Corp. Sample Programs. (C) Copyright IBM Corp. 1998, 2006. All rights reserved.

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

Programming Interface Information

This Application Programming Interfaces (API) publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i5/OS.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:
Advanced 36
Advanced Function Printing
Advanced Peer-to-Peer Networking
AFP
AIX
AS/400
COBOL/400
CUA
DB2
DB2 Universal Database
Distributed Relational Database Architecture
Domino
DPI
DRDA
eServer
GDDM
IBM
Integrated Language Environment
Intelligent Printer Data Stream
IPDS
i5/OS
iSeries
Lotus Notes
MVS
Netfinity
Net.Data
NetView
Notes
OfficeVision
Operating System/2
Operating System/400
OS/2
OS/400
PartnerWorld
PowerPC
PrintManager
Print Services Facility
RISC System/6000
RPG/400
RS/6000
SAA
SecureWay
System/36
System/370
System/38
System/390
VisualAge
WebSphere
xSeries

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.

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

Other company, product, and 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